Journal of Artificial Intelligence Research 30 (2007) 525-564 Submitted 1/07; published 12/07

A Framework for Kernel-Based Multi-Category Classification

Simon I. Hill sih22@eng.cam.ac.uk
Department of Engineering,
University of Cambridge,
Cambridge, UK

Arnaud Doucet arnaud@cs.ubc.ca
Depts. of Statistics and Computer Science
University of British Columbia,
Vancouver, Canada

Abstract

A geometric framework for understanding multi-category classification is introduced, through which many existing 'all-together' algorithms can be understood. The structure enables parsimonious optimisation, through a direct extension of the binary methodology. The focus is on Support Vector Classification, with parallels drawn to related methods.

The ability of the framework to compare algorithms is illustrated by a brief discussion of Fisher consistency. Its utility in improving understanding of multi-category analysis is demonstrated through a derivation of improved generalisation bounds.

It is also described how this architecture provides insights regarding how to further improve on the speed of existing multi-category classification algorithms. An initial example of how this might be achieved is developed in the formulation of a straightforward multi-category Sequential Minimal Optimisation algorithm. Proof-of-concept experimental results have shown that this, combined with the mapping of pairwise results, is comparable with benchmark optimisation speeds.

1. Introduction

The problem of extending classification methods from the standard dichotomous framework to a more general *'polychotomous'* arrangement is one which has been considered by a number of authors. Essentially, the task is to learn from some training data how best to assign one of M possible classes to subsequent input data, where M is known beforehand.

The key contribution of this work is to introduce an overarching framework for understanding multi-category kernel-based classification methods. In particular this is a framework which makes the assumptions and constructions used in individual approaches clear. As a result it enables the operation of most existing multi-category methods to be transparently compared and contrasted in an intuitive and consistent manner. Further, the insight afforded by the architecture suggests ways of developing more efficient algorithms and of bringing together the best of existing techniques.

The central idea behind this approach is to introduce an $(M-1)$-dimensional space which is divided into M class-specific regions. The aim is to learn a $(M-1)$-dimensional function $\mathbf{f}(\cdot)$ which lies in the class region corresponding to the class of its argument. As will be shown this is a straightforward generalisation of the $M=2$ case, in which the two class-specific regions are $f(\cdot) \geq 0$ and $f(\cdot) < 0$. Indeed, in this framework, unlike many other approaches, the binary case is not treated as a special case.

Discussion of this is done primarily in a Support Vector Classification (SVC) context initially, and then extended to other methodologies. The geometric structure employed is introduced in more detail in Section 2, together with a derivation of the optimisation problem, which is shown to be a generalisation of the standard 'all-together' optimisation problems overviewed by Hsu and Lin (2002). This is discussed along with a review of existing Support Vector (SV) multi-category methods in Section 3.

Following this we consider overall algorithm performance with Section 4 discussing Fisher consistency, and Section 5 looking at generalisation bounds. Section 6 then discusses other methodologies, in particular ν-Support Vector Classification (ν-SVC), Least Squares Support Vector Classification (LS-SVC), Lagrangian Support Vector Classification (LSVC), Proximal Support Vector Classification (PSVC), and Bayes Point Machines (BPM). This is followed by a return to the SVC problem and a Sequential Minimal Optimisation (SMO) algorithm is derived in Section 7. Issues related to the details of how best to implement the SMO algorithm (*e.g.* point selection) are discussed, as are options for improving the speed of convergence. These are implemented for several examples in Section 8, in an initial experimental exercise.

2. Setting up the Multi-Category Problem

In this Section the key geometric construction will be presented, as will mechanisms for using this to formulate an optimisation problem. Finally, extensions to the generic structure will be described. The basic construction is described in Subsection 2.1. Following this, Subsection 2.2 describes example empirical SV loss cases, Subsection 2.3 discusses how relative class knowledge can be incorporated and Subsection 2.4 details an overview of the derivation of the SV optimisation problem.

2.1 The Geometric Construction

In the binary classification case, class determination of some input from the set $\mathcal{X}$ is often performed by considering the sign of an underlying real-valued function $f : \mathcal{X} \to \mathbb{R}$ (Vapnik, 1998, for example). In progressing to the M-class case, the underlying vector-valued function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^{M-1}$ will be found, where $\mathbf{f} = \begin{bmatrix} f_1 & \ldots & f_{M-1} \end{bmatrix}^T$. The basic idea behind the use of an $(M-1)$-dimensional space is to be able to introduce M equally separable class-target vectors. The class of input $\mathbf{x}$ will be determined by identifying that class-target vector to which $\mathbf{f}(\mathbf{x})$ is closest.

This can be seen to effectively be what takes place in binary SV classification, where classes, denoted A and B, have class targets $y(A) = -1$ and $y(B) = +1$. Consider now that a third class, C, is a possibility. A one-dimensional numerical label is insufficient for the classes to be equidistant, and in the case that little is known about the relationship between the classes then the logical arrangement would be to compare every class to every other in an equivalent way. In order to do this then class targets must be equidistant in some sense.

A two-dimensional arrangement as illustrated in Figure 1 allows this. Here the class-target vectors are

$$\mathbf{y}(A) = \begin{bmatrix} -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}^T, \quad \mathbf{y}(B) = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}^T, \quad \mathbf{y}(C) = \begin{bmatrix} 0 & 1 \end{bmatrix}^T. \tag{1}$$

where $\|\mathbf{y}(\vartheta)\| = 1$[1] for all classes $\vartheta \in \Theta$ (with $\Theta = \{A, B, \dots\}$ denoting the set of possible classes) as this improves tractability later. These are example class-target vectors, however,

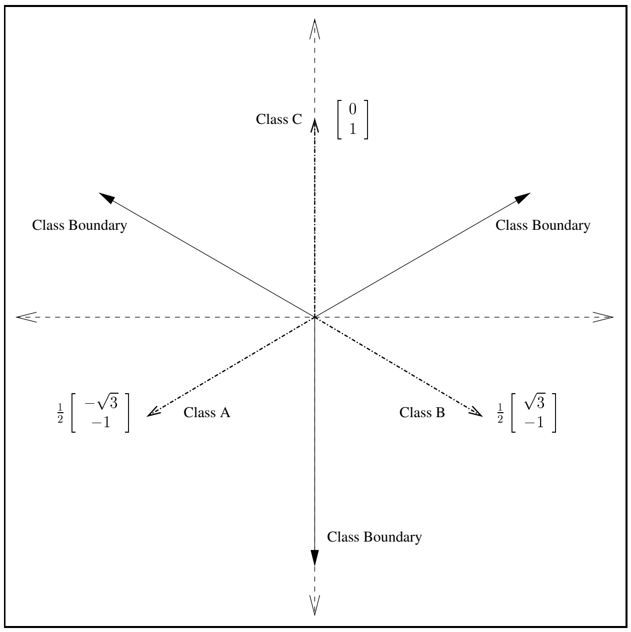


Figure 1: Possible class labels for classification into three. *The class-target vectors corresponding to classes A, B and C are shown. The class boundaries are given by solid lines.*

in general it is important to understand that the optimisation methods which will be described are applicable regardless of their rotation. Indeed, although the apparent Cartesian coordinate asymmetry may not appear intuitive, the important consideration is the relative positioning of class-target vectors with respect to each other. The optimisation procedure has no dependence on any particular orientation. This will be proven for SV methods as part of the derivation of the optimisation process in Section 2.4.

The same approach to that described for $M = 3$ is taken when considering larger values of M. While typically $M = 3$ will be used in this work as an example case, extensions to higher values of M follow without further consideration. An example of how target vectors might easily be found in higher dimensions is discussed by Hill and Doucet (2005).

1. Note that in this work $\|\cdot\|$ denotes the 2-norm of a vector, *i.e.* $\|\mathbf{y}\| = \sqrt{y_1^2 + \cdots + y_{M-1}^2}$ and, further, normalisation will imply $\frac{\mathbf{y}}{\|\mathbf{y}\|}$

2.2 SV Empirical Multi-Category Loss

In setting up the classification process, each class is assigned a subset of the $(M-1)$-dimensional output space. In particular, in the most straightforward approach, these subsets are the Voronoi regions associated with the class targets. As a result, class boundaries can be found by forming hyperplanes between class regions which consist of all points equidistant from the two relevant class targets. For an input $\mathbf{x}$, the classifier output is given by the function h which is found by observing in which of the regions $\mathbf{f}(\mathbf{x})$ lies *i.e.*

$$h(\mathbf{x}) = \text{The class associated with the region in which } \mathbf{f}(\mathbf{x}) \text{ lies.} \tag{2}$$

In describing empirical loss the vectors perpendicular to the hyperplane dividing the region between $\mathbf{y}(A)$ and $\mathbf{y}(B)$ will typically be used[2]. Define

$$\mathbf{v}_A(B) = \frac{\mathbf{y}(B) - \mathbf{y}(A)}{\|\mathbf{y}(B) - \mathbf{y}(A)\|} \tag{3}$$

These vectors are illustrated for class C in Figure 2 in which a margin ε has been introduced and defined as $\varepsilon = \mathbf{y}^T(\vartheta)\mathbf{v}_\theta(\vartheta)$ for all $\theta, \vartheta \in \{A, B, C\}$ and $\theta \neq \vartheta$. Note that here the dependency on θ, ϑ is not explicitly noted when referring to ε as it is constant. Discussions of when this might not be the case are presented later. This definition of vectors $\mathbf{v}$ is used as the aim will be to measure distance in a direction perpendicular to the class boundaries and this can be done through an inner product with the relevant vector $\mathbf{v}$.

This margin is used for all cases in finding the empirical loss. While there are several different ways to combine individual loss components, the fundamental starting point is that illustrated in Figure 2. Here a training point $\mathbf{x}$ with class C has $\mathbf{f}(\mathbf{x})$ which falls outside the required region. This is penalised by $\left(\varepsilon - \mathbf{v}_B^T(C)\mathbf{f}(\mathbf{x})\right)$ in an analogous way to the binary SV classification empirical loss of $(1 - yf(\mathbf{x}))$. Indeed in the binary case $v_B(A) = y(A)$ and $v_A(B) = y(B)$ and $\varepsilon = 1$. As a further parallel, just as there is a region of zero loss in the binary case when $y \cdot f(\mathbf{x}) > 1$, so too is there a region of zero loss here, above the dotted lines.

Consider now that training data $\{(\mathbf{x}_i, \vartheta_i) : i \in \{1, \ldots, N\}\}$ is to be used to learn how best to classify some new input $\mathbf{x}$. Denote the indicator function by $\mathbb{I}(\cdot)$; the empirical loss for a polychotomous classification problem given by Allwein, Schapire, and Singer (2001), Crammer and Singer (2001a), and Lee et al. (2001, 2004) is then,

$$\ell_{EMP} = \sum_{i=1}^{N} \mathbb{I}(h(\mathbf{x}_i) \neq \vartheta_i), \tag{4}$$

namely the number of misclassified training samples. As with dichotomous SV techniques, some loss will be used which bounds ℓ_{EMP}, thus generating a straightforward optimisation problem.

In setting up multi-category SV classification, this is an approach used by many different authors, however their exact empirical loss functions have differed. The most prevalent can be understood within the framework of Figures 1 and 2, four of these are illustrated in Figure 3 for an object of class C. These four loss functions involve either adding together

2. An exception to this is the case presented by Lee, Lin, and Wahba (2001, 2004), as discussed by Hill and Doucet (2005, App. C).

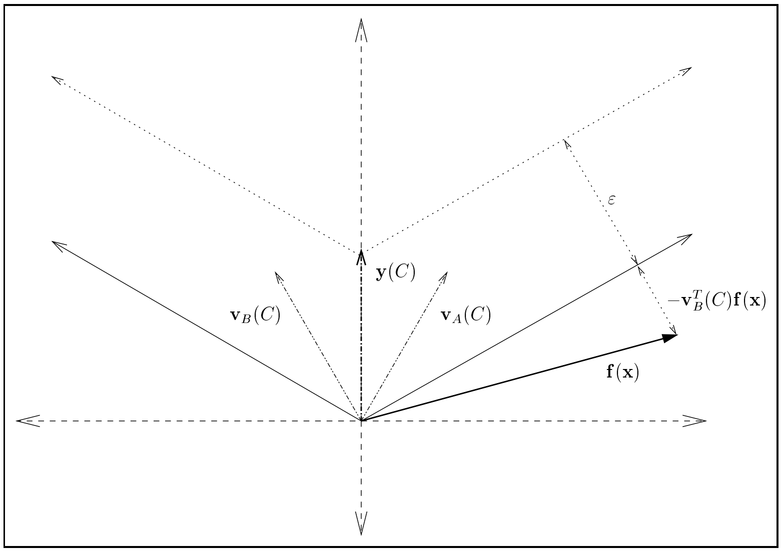


Figure 2: Elements involved in determining empirical loss associated with a training sample of class C. *Note that the unlabelled solid lines are the class boundaries, the region above the dotted line is the region of zero loss for training samples of class* C*.*

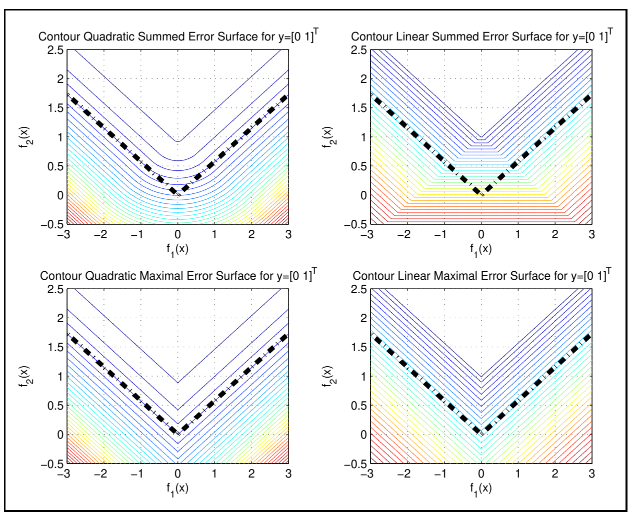


Figure 3: Four possible loss functions for the three class problem (see Figure 1). *The loss functions are shown with respect to the target vector* $\mathbf{y} = [0\ 1]^T$*. Traditional additive losses are shown at top, (see equations* (6) *and* (5)*), possible variants following proposals by Crammer and Singer (2001a) at bottom (see equations* (8) *and* (7)*). In all cases the class boundary is shown by a dot-dash line.*

all margin infringements, or taking the largest such infringement. Both linear and quadratic versions of these two options are illustrated. Algebraically, the summed loss for training point i can be expressed,

$$\ell_{SL,i} = \sum_{\theta \in (\Theta - \vartheta_i)} \max\left(\left(\varepsilon - \mathbf{f}^T(\mathbf{x}_i)\mathbf{v}_\theta(\vartheta_i)\right), 0\right) \tag{5}$$

$$\ell_{SQ,i} = \sum_{\theta \in (\Theta - \vartheta_i)} \left[\max\left(\left(\varepsilon - \mathbf{f}^T(\mathbf{x}_i)\mathbf{v}_\theta(\vartheta_i)\right), 0\right)\right]^2 \tag{6}$$

where SL stands for summed linear loss and SQ for summed quadratic loss. These are the top two Subfigures in Figure 3. Using the same notation, the maximal loss for training point i can be expressed,

$$\ell_{ML,i} = \max_{\theta \in (\Theta - \vartheta_i)} \left\{\max\left(\varepsilon - \mathbf{f}^T(\mathbf{x}_i)\mathbf{v}_\theta(\vartheta_i), 0\right)\right\} \tag{7}$$

$$\ell_{MQ,i} = \max_{\theta \in (\Theta - \vartheta_i)} \left\{\left[\max\left(\varepsilon - \mathbf{f}^T(\mathbf{x}_i)\mathbf{v}_\theta(\vartheta_i), 0\right)\right]^2\right\} \tag{8}$$

where ML stands for maximal linear and MQ for maximal quadratic. These are the bottom two Subfigures in Figure 3. From these expressions it is apparent that the ith summand of the empirical loss (equation (4)) is bound by $\frac{1}{\varepsilon^2} \times \ell_{SQ,i}$, $\frac{1}{\varepsilon^2} \times \ell_{MQ,i}$, $\frac{1}{\varepsilon} \times \ell_{SL,i}$ and $\frac{1}{\varepsilon} \times \ell_{ML,i}$. While all of these loss arrangements can be cast in a transparent way into a SV framework, in this work only $\ell_{SL,i}$ will initially be focussed on, as it has been most commonly adopted, albeit implicitly, in previous contributions. $\ell_{SQ,i}$ will be discussed with respect to LSVC in Subsection 6.3.

In terms of the practioner's preferred approach, however, clearly the choice must be in line with the underlying probabilistic model of the data. It seems unlikely that there will be one best choice for all implementations. In the case that the practioner has no particular idea about a model and just wishes to use some methodology to 'get a feel' for the data, then presumably it is optimal to use the most computationally efficient approach as often these approaches will converge to very similar results. To this end the approach outlined in this paper is of interest as it describes methods which can potentially be used to speed all loss cases.

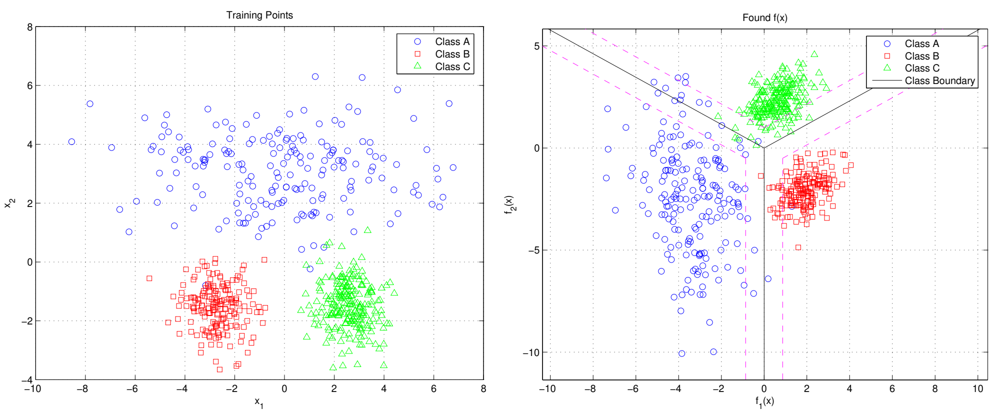
2.3 Relative Class Knowledge

While the framework developed has been based on the assumption that all classes are to be treated equally, this may not be desirable in some cases. There may be some prior knowledge suggesting that some classes are, in some sense, closer to each other, and thus more likely to be mistaken for each other. There may also be some reason for preferring to err on the side of choosing one class over the others or over another at the cost of overall accuracy.

A classical example of deeming it more important to choose one class over another comes from the binary case of detection by radar. In military combat it is clearly extremely important to detect incoming missiles or planes. As a result it is understandable that a classification algorithm may be set up to return many more *false positives* than *false negatives*. Hence errors made when classing enemy weaponry as unimportant are far more heavily penalised than errors made in classifying nonthreatening phenomena as weaponry.

There are two ways to introduce relative class knowledge in the framework presented. The first of these is the traditional method of error weighting, as introduced to the 'all-together' SV framework by Lee et al. (2001). In this solution each different type of misclassification (*e.g.* classifying an input as θ instead of ϑ) has its error weighted by some amount; $D_\theta(\vartheta)$.

This approach of incorporating weights could equivalently be viewed as varying the length of the vectors $\mathbf{v}$, *i.e.* $\mathbf{v}_\theta(\vartheta) \to D_\theta(\vartheta)\mathbf{v}_\theta(\vartheta)$. An alternative, and possibly complementary, approach is to allocate to each class an unequal volume in the $(M-1)$-dimensional output space. This can be enabled by varying the angle between the class boundaries and hence the orientation of the vectors $\mathbf{v}$, *i.e.* $\mathbf{v}_\theta(\vartheta) \to \mathbf{R}_\theta(\vartheta)\mathbf{v}_\theta(\vartheta)$ where $\mathbf{R}_\theta(\vartheta)$ is some rotation matrix. In doing this it may also be useful to incorporate a set of variable ε values which, for some class ϑ are denoted $\{\varepsilon_\theta(\vartheta) : \theta \in (\Theta - \vartheta)\}$, that is $\varepsilon_\theta(\vartheta)$ is the size of the margin on the ϑ side of the (ϑ, θ) boundary. Clearly the greater the volume allocated to the class the more diverse the input vectors can be which are mapped to it.



(a) Classes in feature space

(b) Output result

Figure 4: A simple example illustrating a potential case for differently sized class areas. *In this arrangement the target area for class A could be increased.*

Unfortunately it is not obvious how to construct a principled approach to determining these different volumes. The key issue is the region of support that each class has in the feature space. For instance in the case illustrated in Figure 4 it is not possible to find a linear projection from the feature space which will separate the classes into the standard class regions. However, by changing the class region sizes such a projection would be possible. This may have the advantage of avoiding a more complicated feature space (possibly of higher dimension).

2.4 Derivation of the SVC Optimisation Problem

Standard SV mappings of inputs to a higher dimensional *feature* space, $\Phi : \mathcal{X} \to F$ are used in order to estimate the $(M-1)$-dimensional function $\mathbf{f}(\cdot)$. The mth element of $\mathbf{f}(\cdot)$ is a linear function in this feature space, characterised by weight vector $\mathbf{w}_m$ and offset b_m. To summarise,

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \langle \Phi(\mathbf{x}), \mathbf{w}_1 \rangle_F \\ \langle \Phi(\mathbf{x}), \mathbf{w}_2 \rangle_F \\ \vdots \\ \left\langle \Phi(\mathbf{x}), \mathbf{w}_{(M-1)} \right\rangle_F \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{(M-1)} \end{bmatrix} = \boldsymbol{\psi}(\mathbf{x}) + \mathbf{b}. \tag{9}$$

It is important to realise that, although some class separation is achieved by each component, $f_m(\cdot)$, accurate classification can only really be accomplished through the use of all elements, together.

The optimisation problem which follows from the discussion in the previous Subsections can be written (in standard SV form) as,

$$\begin{aligned} &\text{Minimise} \quad \left[\frac{1}{2} \sum_{m=1}^{M-1} \|\mathbf{w}_m\|_F^2 + C \sum_{i=1}^{N} \sum_{\theta \in (\Theta - \vartheta_i)} D_\theta(\vartheta_i) \xi_{i,\theta} \right] \\ &\text{Subject to} \quad \begin{cases} \sum_{m=1}^{M-1} \left(\langle \Phi(\mathbf{x}_i), \mathbf{w}_m \rangle_F + b_m \right) v_{\theta,m}(\vartheta_i) \geq \varepsilon_\theta(\vartheta_i) - \xi_{i,\theta}, \text{ for } i = 1, ..., N, \ \theta \in (\Theta - \vartheta_i) \\ \xi_{i,\theta} \geq 0, \text{ for } i = 1, ..., N, \ \theta \in (\Theta - \vartheta_i) \end{cases} \end{aligned} \tag{10}$$

where the slack variable $\xi_{i,\theta}$ quantifies the empirical loss involved in mistaking the class of point $\mathbf{x}_i$ (which is ϑ_i) for $\theta (\neq \vartheta_i)$. C quantifies the trade-off between regularisation (introduced by $\|\mathbf{w}_m\|_F^2$) and this empirical loss, and $v_{\theta,m}(\vartheta)$ is the mth element of $\mathbf{v}_\theta(\vartheta)$. Framing equation (10) in terms of a Lagrangian gives,

$$\begin{aligned} L = &\frac{1}{2} \sum_{m=1}^{M-1} \|\mathbf{w}_m\|_F^2 + C \sum_{i=1}^{N} \sum_{\theta \in (\Theta - \vartheta_i)} D_\theta(\vartheta_i) \xi_{i,\theta} - \sum_{i=1}^{N} \sum_{\theta \in (\Theta - \vartheta_i)} r_{i,\theta} \xi_{i,\theta} \\ &- \sum_{i=1}^{N} \sum_{\theta \in (\Theta - \vartheta_i)} \alpha_{i,\theta} \left(\sum_{m=1}^{M-1} \left(\langle \Phi(\mathbf{x}_i), \mathbf{w}_m \rangle_F + b_m \right) v_{\theta,m}(\vartheta_i) - \varepsilon_\theta(\vartheta_i) + \xi_{i,\theta} \right) \end{aligned} \tag{11}$$

where $\{\alpha_{i,\theta}, r_{i,\theta} : i \in (1, \ldots, N), \theta \in (\Theta - \vartheta_i)\}$ are Lagrangian multipliers. It is standard in SV methodology to find the optimal solution to this by first finding the Wolfe dual, and then maximising with respect to the dual variables, namely the Lagrangian multipliers (Cristianini & Shawe-Taylor, 2000; Vapnik, 1998, for example). First let $\mathbf{V}(\vartheta)$ denote a $(M-1) \times (M-1)$ matrix with columns given by the vectors $\mathbf{v}_\theta(\vartheta)$,

$$\mathbf{V}(\vartheta) = \begin{bmatrix} \mathbf{v}_A(\vartheta) & \mathbf{v}_B(\vartheta) & \ldots & \mathbf{v}_{\theta \neq \vartheta}(\vartheta) & \ldots \end{bmatrix} \tag{12}$$

and represent the mth row of $\mathbf{V}(\vartheta)$ by $\mathbf{v}_m^{*T}(\vartheta)$.

Lemma 1 *The dual to the Lagrangian presented in equation* (11) *is,*

$$L_D = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{\alpha}_i^T \mathbf{V}^T(\vartheta_i) \mathbf{V}(\vartheta_j) \boldsymbol{\alpha}_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{N} \boldsymbol{\alpha}_i^T \boldsymbol{\varepsilon}(\vartheta_i) \tag{13}$$

where,

$$\boldsymbol{\alpha}_i = \begin{bmatrix} \alpha_{i,A} & \alpha_{i,B} & \dots & \alpha_{i,\theta\neq\vartheta_i} & \dots \end{bmatrix}^T \tag{14}$$

$$\boldsymbol{\varepsilon}(\vartheta_i) = \begin{bmatrix} \varepsilon_A(\vartheta_i) & \varepsilon_B(\vartheta_i) & \dots & \varepsilon_{\theta\neq\vartheta_i}(\vartheta_i) & \dots \end{bmatrix}^T \tag{15}$$

and the kernel function has been denoted $K(\mathbf{x}_i, \mathbf{x}_j) = \langle\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)\rangle_F$*. The derivation of equation* (13) *also introduces the constraints that,*

$$CD_\theta(\vartheta_i) \geq \alpha_{i,\theta} \geq 0, \ \forall i, \theta \in (\Theta - \vartheta_i) \tag{16}$$

$$\sum_{i=1}^{N} \mathbf{V}(\vartheta_i)\boldsymbol{\alpha}_i = \mathbf{0}. \tag{17}$$

The derivation of this is presented in a technical report by the authors Hill and Doucet (2005, App. A).

It also remains to confirm that this optimisation problem has a unique maximum, that is that the problem is unimodal. This will be the case if it can be shown that the quadratic term in equation (13) is effectively equivalent to a quadratic expression involving a positive definite matrix. This is the case, as shown by Hill and Doucet (2005, App. B).

A final issue to consider is that of rotational invariance to the structuring of the problem — as initially raised in Subsection 2.1. Note that the only influence of rotational orientation in equation (13) is through the summation term $\boldsymbol{\alpha}_i^T\mathbf{V}^T(\vartheta_i)\mathbf{V}(\vartheta_j)\boldsymbol{\alpha}_j K(\mathbf{x}_i, \mathbf{x}_j)$. Consider now that the chosen orientation is rotated in some way as described by a rotation matrix $\mathbf{R}$, this quadratic term then becomes,

$$\boldsymbol{\alpha}_i^T\mathbf{V}^T(\vartheta_i)\mathbf{R}^T\mathbf{R}\mathbf{V}(\vartheta_j)\boldsymbol{\alpha}_j K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\alpha}_i^T\mathbf{V}^T(\vartheta_i)\mathbf{V}(\vartheta_j)\boldsymbol{\alpha}_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{18}$$

due to the fact that rotation matrices are orthonormal. There is one further aspect that should be considered, namely the constraints in equation (17), however it is clear that these will not be affected by rotation either. Hence the optimisation problem is rotationally invariant.

A related issue is that the geometric structure implicitly introduces ordinal regression along the $(M-1)$ axes. That is, when looking for example, at the three-class case illustrated in Figure 1 there are essentially two real valued outputs *i.e.* $f_1(\cdot)$ and $f_2(\cdot)$. Now, along any horizontal or vertical line for which one of these is held constant, the other is outputing a value, and the region into which this value falls determines the class assignment. This gives the impression of ordinal regression as using ranges of a single value output to determine between more than two classes is the essence of that approach.

This raises two questions — is the methodology presented subject to the same problems as ordinal regression? And, when looking at the structure in this way, does the fact that it can potentially be arranged quite asymmetrically, and appears arbitrary cause concern?

The answer to both questions is 'No'. The very aim of structuring the ouput space in this way has been to avoid the situation encountered in ordinal regression in which classes are not all equivalently compared against each other. Furthermore, it should be clear from the rotational invariance of the structure, that the particular orientation chosen is not going to affect the optimisation problem in any way whatsoever.

Note that with the introduced terminology then the function $\mathbf{f}(\cdot)$ can be expressed,

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{V}(\vartheta_i)\boldsymbol{\alpha}_i K(\mathbf{x}, \mathbf{x}_i) + \mathbf{b}. \tag{19}$$

This is clearly a very natural extension of the binary framework, a comparison with previous similar contributions forms the next Section. The offset, $\mathbf{b}$ can be determined through realising that the non-extremal Lagrangian coefficients $\alpha_{i,\theta}$ lie on the edge of the zero-loss region — this is analogous to finding b in the two-class case.

3. Discussion of Previous SV Approaches

There are three main methods for applying binary classification techniques to the more generic multi-category problem. These are the *one-against-all* method, the *pairwise coupling* method and the method of *Error Correcting Output Codes (ECOCs)*. As will be discussed here, all have been applied in conjunction with SV methods. An extensive literature review forms a large part of work by Rifkin and Klautau (2004); by contrast, in this Section the various methods are discussed with respect to the approach described in Section 2.

Essentially, while the one-against-all and pairwise methods can be made to work well, they invariably require a heuristic component to resolve issues associated with combining results. Many authors have tried to overcome this by framing the problem as a single optimisation, as is also done in Section 2, however these approaches are substantially slower to converge. A key contribution of this work is to demonstrate that a consistent[3] result can be obtained by mapping pairwise results in an *ad hoc* way into the framework of Figure 1 and 'fine-tuning' the result, to a consistent final optimum. This provides a combination of fast training and consistency.

Contributions such as that by Rifkin and Klautau (2004) argue that one-against-all and pairwise methods can be made to perform practically as well as other methods. This may well be the case, depending on the implicit model behind the heuristics involved and should come as no surprise. Indeed, for many quick black-box implementations this approach may well be optimal.

Often however it is desirable to have a clear understanding of the optimisation process and a significant contribution of the framework presented is that within it many single optimisation SV methods can be understood and compared directly. Further, it is a framework in which multi-category versions of other algorithms can be formulated and understood in a consistent way, as will be discussed in Section 6. Finally, the very fact that so many different efforts have been made to find a method involving a single optimisation which is competitive in terms of speed is in itself evidence of a desire by the research community to overcome heuristic solutions.

In this Section the one-against-all method is reviewed in Subsection 3.1, pairwise coupling in Subsection 3.2 and ECOCs in Subsection 3.3. Efforts to develop single optimisation

3. Here, and in subsequent usage, we use 'consistent' and 'consistency' to refer to the fact that while some approaches can be quite ambiguous in exactly which class to output, hence the need for a heuristic to choose, an 'all-together' solution should not suffer from this *i.e.* the results are consistent with each other. Fisher consistency is discussed in Section 4.

approaches, known as 'all-together' methods are discussed in Subsection 3.4. How they relate to the framework presented is also clarified therein.

3.1 The One-Against-All Method

The one-against-all method has received the most attention from the SV community, and was also the earliest approach considered. The idea is to generate M classifiers. Of these, classifier i determines whether or not an input belongs to class i. An obvious stumbling-block is the case that more than one classifier determines that a particular input should belong to its class of interest. Hence it is important to have in place some technique for either avoiding or resolving this problem.

Early implementations (Schölkopf, Burges, & Vapnik, 1995; Blanz, Schölkopf, Bülthoff, Burges, Vapnik, & Vetter, 1996) used the underlying real-valued output, choosing the highest such output to indicate the strongest 'likelihood' of class membership. A variant on this was introduced by Mayoraz and Alpaydın (1999) in an attempt to make these outputs more reliably comparable; see also Wang, Vuurpijl, and Schomaker (2000).

As an aside, note that a function $\mathbf{f}(\cdot)$ found in the framework proposed in Section 2 can be used to produce a one-against-all classifier. To see this consider the function $f_\theta(\cdot) = \mathbf{y}^T(\theta)\mathbf{f}(\cdot)$ where $\mathbf{y}(\theta)$ is the class-target vector for class θ. A new input $\mathbf{x}$ would then be classed θ if $f_\theta(\mathbf{x})$ is the largest such scalar function.

3.2 The Pairwise Coupling Method

The pairwise coupling method involves finding $\frac{M(M-1)}{2}$ different classifiers, each of which compares one class against another. Given an input, the output class is decided through a vote or some similar method. One particular problem to be addressed is the circular one in which, for example, the AB classifier chooses class A, the AC classifier class C and the BC classifier class B.

A few authors (Hastie & Tibshirani, 1998; Platt, Cristianini, & Shawe-Taylor, 2000; Fürnkranz, 2002) have proposed heuristics for resolving any such problems, however the most substantive purely SV approach seems to be that by Kreßel (1999). This technique considers a classifier vote and, in the case of a tie, the real-valued outputs are referred to. This has the downside that the $\frac{M(M-1)}{2}$ classifiers are found independently, and so such a comparison is not always meaningful. Nonetheless, as is shown by Kreßel and supported by Allwein et al. (2001), Hsu and Lin (2002) and Rifkin and Klautau (2004), this appears to be an effective methodology and faster than 'all-together' methods by a considerable margin. An alternative approach is presented by Allwein et al. (2001) in their work unifying pairwise, one-against-all and class codes — this is discussed in Subsection 3.3.

It is interesting to note that a function $\mathbf{f}(\cdot)$ found in the framework proposed in Section 2 can be used to find a classifier between two classes A and B. To see this consider the function $f_{AB}(\cdot) = \mathbf{v}_A^T(B)\mathbf{f}(\cdot)$ where $\mathbf{v}_A(B)$ is as defined in equation (3). A new input $\mathbf{x}$ would then be classed A if $f_{AB}(\mathbf{x}) \leq 0$ and B otherwise. Results such as these will be used in Section 7 to construct the 'best of both worlds' approach.

3.3 Class Codes

The underlying idea behind ECOCs is to assign to each class a particular binary code and then to train individual classifiers to identify each bit in the code (Sejnowski & Rosenberg, 1987; Dietterich & Bakiri, 1995). Eight classes can, for example, be each assigned a 3-bit code ($[-1,-1,-1]$, $[-1,-1,+1]$, etc.). In general at least $\lceil \log_2 M \rceil$ bits are required. Dietterich and Bakiri (1995) propose using more than M bits in order that small errors can be corrected.

ECOC SV methods have been described by Kindermann, Leopold, and Paaß (2000) and Rennie and Rifkin (2001). Minimum length code methods have been presented by Sebald (2000), Sebald and Bucklew (2001) and by Suykens and Vandewalle (1999). However these implementations often have the problem that classes are treated inconsistently. This is due to the fact that such codes will have smaller Hamming distances between some classes than others. Their approach becomes very much like that of utilising *Ordinal Regression* (see, for example, Crammer and Singer (2001b) or Herbrich, Graepel, and Obermayer (2000b) for more, in a SV context) to perform the classification and performance becomes dependent on the ordering of the labels (Weston, 1999). Essentially ordinal regression performs classification based on the region of $\mathbb{R}$ in which some scalar output lies. For instance, a class of A may be assigned to some input $\mathbf{x}$ if $f(\mathbf{x}) \in [a, b)$, class B if $f(\mathbf{x}) \in [b, c)$, and so on.

To see the parallel between this and coding approaches consider the four-class case with minimum length codes, such that,

$$\begin{array}{ll} \mathbf{y}(A) = \begin{bmatrix} -1 & -1 \end{bmatrix} & \mathbf{y}(B) = \begin{bmatrix} -1 & +1 \end{bmatrix} \\ \mathbf{y}(C) = \begin{bmatrix} +1 & -1 \end{bmatrix} & \mathbf{y}(D) = \begin{bmatrix} +1 & +1 \end{bmatrix}. \end{array}$$

Two functions f_1 and f_2 need to be found which corresspond to the first and second elements of the codes respectively. However class D is clearly further from class A than from classes B and C. Hence the comparison between classes is again no longer consistent. Although this is less extreme than ordinal regression the main problem that classes are not compared in an equivalent way remains. For this reason, as well as the lack of a computational or accuracy advantage, these methods have not been particularly popular.

Allwein et al. (2001) have shown that pairwise and one-against-all methods can be viewed as special cases of ECOC approaches[4]. Indeed it is only when the code length equals (the one-against-all case) or exceeds M that the inconsistency problem described above can be made to disappear. Even when viewing pairwise and one-against-all approaches as special cases of ECOC Allwein et al. (2001) still must employ a heuristic (in this case code-based) to find the final answer.

3.4 'All-Together' Methods

A consistent result can be obtained by arranging the multi-category problem such that there is a single optimisation to perform. These are described by Hsu and Lin (2002) as 'all-together' methods and a number of authors (Bredensteiner & Bennett, 1999; Crammer

4. An exception to the binary nature of the code in this formulation being the case of pairwise comparison when code word elements take values from $\{-1, 0, 1\}$.

& Singer, 2001a; Guermeur, 2000, 2002; Weston & Watkins, 1999; Weston, 1999; Vapnik, 1998) present a variety of such methods. To see that many of these relate to that described in Section 2, note that their aim is to find M functions $\{f'_\vartheta(\cdot) : \vartheta \in \Theta\}$ such that, $\text{class}(\mathbf{x}) = \arg\max_\vartheta \{f'_\vartheta(\mathbf{x}) : \vartheta \in \Theta\}$. Weston and Watkins (1999) aim to find functions of the form $f'_\vartheta(\cdot) = \langle \Phi(\cdot), \mathbf{w}'_\vartheta \rangle + b'_\vartheta$ by

$$\begin{array}{ll} \text{Minimising} & \dfrac{1}{2}\displaystyle\sum_{\theta\in\Theta} \|\mathbf{w}'_\theta\|_F^2 + C\sum_{i=1}^{N}\sum_{\theta\in(\Theta-\vartheta_i)} \xi'_{i,\theta} \\ \text{Subject to} & \begin{cases} \left\langle \Phi(\mathbf{x}_i), \mathbf{w}'_{\vartheta_i} \right\rangle_F + b_{\vartheta_i} \geq \langle \Phi(\mathbf{x}_i), \mathbf{w}'_\theta \rangle_F + b_\theta + 2 - \xi'_{i,\theta}, \;\; \theta \in (\Theta - \vartheta_i) \\ \xi'_{i,\theta} \geq 0 \end{cases} \end{array} \tag{20}$$

and it has been shown in detail by the authors (Hill & Doucet, 2005, App. C) that this optimisation arrangement is identical to that in Section 2 when $f'_\vartheta(\cdot) = \mathbf{y}^T(\vartheta)\mathbf{f}(\cdot)$ where $\mathbf{f}(\cdot)$ is as introduced in Section 2. Furthermore, the other 'all-together' approaches mentioned, with the exception of Crammer and Singer (2001a), have been shown by Guermeur (2002) to converge to the same solution. However a key problem with these algorithms is that their resulting kernel expressions can be quite complicated in that the framing of the optimisation process leads to convoluted expressions.

As an alternative Crammer and Singer (2001a) propose an 'all-together' one-against-all method with the maximal loss $\ell_{ML,i}$ in equation (7) (see also Figure 3) for which they give a new optimisation algorithm. In their comparative work Hsu and Lin (2002) find it hard to draw definitive conclusions about the Crammer and Singer (2001a) approach in comparison to the others as they note variable performance with regard to optimisation times required. They also suggest algorithmic improvements to the more traditional methods, but eventually conclude that, of available techniques, pairwise coupling methods (see Section 3.2) are much faster and appear more suitable for practical use.

While the standard form of the methodology introduced in Section 2 results in an optimal solution equivalent to other 'all-together' approaches two key points differentiate it. The first is that it has increased flexibility in that it can incorporate the approaches described in Subsection 2.3 without any increased computational effort. The second is that it can easily take advantage of the relatively much faster pairwise methods. This is discussed further in Subsection 7.4.

3.4.1 Another 'All-Together' Approach

Lee et al. (2001, 2004) have presented a unique 'all-together' approach which uses M-length target codes. For classes $\Theta = \{A, B, \dots\}$ these take the form

$$\begin{aligned} \mathbf{y}''(A) &= \left[\begin{array}{cccc} 1 & \frac{-1}{M-1} & \cdots & \frac{-1}{M-1} \end{array}\right] \\ \mathbf{y}''(B) &= \left[\begin{array}{ccccc} \frac{-1}{M-1} & 1 & \frac{-1}{M-1} & \cdots & \frac{-1}{M-1} \end{array}\right] \end{aligned}$$

and so on. The resulting optimisation problem posed is to

$$\begin{aligned}&\text{Minimise } \frac{1}{2}\sum_{\theta\in\Theta}\|\mathbf{w}''_\theta\|_F^2 + C\sum_{i=1}^{N}\sum_{\theta\in(\Theta-\vartheta_i)}\xi''_{i,\theta}\\ &\text{Subject to } \left\{\begin{array}{l}\langle\Phi(\mathbf{x}_i),\mathbf{w}''_\theta\rangle_F + b''_\theta - y''_i(\theta) \le \xi''_{i,\theta}\\ \sum_{\theta\in\Theta}\left(\langle\Phi(\mathbf{x}_i),\mathbf{w}''_\theta\rangle_F + b''_\theta\right) = 0\\ \xi''_{i,\theta}\ge 0\end{array}\right.\end{aligned} \tag{21}$$

where output is given by $\mathbf{f}''(\cdot) = \left[\begin{array}{ccc}\left(\langle\Phi(\cdot),\mathbf{w}''_A\rangle_F + b''_A\right) & \left(\langle\Phi(\cdot),\mathbf{w}''_B\rangle_F + b''_B\right) & \dots\end{array}\right]^T = \mathbf{W}''\Phi(\cdot) + \mathbf{b}''$ with $\mathbf{W}'' = \left[\begin{array}{ccc}\mathbf{w}''_A & \mathbf{w}''_B & \dots\end{array}\right]^T$, and the class membership determined by observing which element of $\mathbf{f}''(\cdot)$ is maximal. This approach can be understood in the framework of Section 2 when, similarly to Subsection 3.4 $f''_\vartheta(\cdot) = \mathbf{y}^T(\vartheta)\mathbf{f}(\cdot)$ where $\mathbf{f}(\cdot)$ is as introduced in Section 2. This has been discussed further by Hill and Doucet (2005, App. C), where it is shown that setting $\mathbf{v}_A(B) = -\mathbf{y}(A)$ and $\varepsilon = \frac{1}{M-1}$ causes the optimisation problem in equation (21) to be the same as the generic approach in equation (10).

4. A Brief Look at Fisher Consistency

The Lee et al. (2001, 2004) (Subsubsection 3.4.1) approach has the key feature that it is Fisher consistent[5]. This has recently been discussed in the context of multicategory classification by Tewari and Bartlett (2007), Zhang (2004a, 2004b). In this Section we simply aim to show how some of the key results of these authors can be understood in the framework presented in Section 2.

In considering Fisher consistency in the multicategory case we first define the vector

$$\mathbf{p}(\mathbf{x}) = \left[\begin{array}{ccc}p_A(\mathbf{x}) & p_B(\mathbf{x}) & \dots\end{array}\right]^T = \left[\begin{array}{ccc}P(\vartheta=A|\mathbf{x}) & P(\vartheta=B|\mathbf{x}) & \dots\end{array}\right]^T \tag{22}$$

this is analogous to the vector $\mathbf{p}$ of Tewari and Bartlett (2007), or the vector $\mathbf{P}(\cdot|X)$ of Zhang (2004b, Eqn.(7)). We also express the empirical loss function, by $\ell(\vartheta,\mathbf{f}(\mathbf{x})) = \sum_{\theta\in(\Theta-\vartheta)}\max\left(\left[\varepsilon - \mathbf{v}_\theta^T(\vartheta)\mathbf{f}(\mathbf{x})\right],0\right)$, *cf.* equation (5), and define the vector $\boldsymbol{\ell}$ by,

$$\boldsymbol{\ell}(\mathbf{f}(\mathbf{x})) = \left[\begin{array}{ccc}\ell(A,\mathbf{f}(\mathbf{x})) & \ell(B,\mathbf{f}(\mathbf{x})) & \dots\end{array}\right]^T. \tag{23}$$

With this notation then the 'ℓ-risk', in the terminology of Tewari and Bartlett (2007) is given by,

$$\begin{aligned}\mathbb{E}_{\mathcal{X}\times\Theta}\left[\ell(\vartheta,\mathbf{f}(\mathbf{x}))\right] &= \mathbb{E}_{\mathcal{X}}\left[\mathbb{E}_{\Theta|\mathcal{X}}\left[\ell(\vartheta,\mathbf{f}(\mathbf{x}))\right]\right]\\ &= \mathbb{E}_{\mathcal{X}}\left[\mathbf{p}^T(\mathbf{x})\boldsymbol{\ell}(\mathbf{f}(\mathbf{x}))\right]\end{aligned} \tag{24}$$

and the optimal classification rule is to choose class $\vartheta^* = \arg\max_\theta\left[p_\theta(\mathbf{x})\right]$. A Fisher consistent multicategory classifier will have $\mathbf{g}(\mathbf{x}) = \mathbf{Y}^T\mathbf{f}(\mathbf{x})$, *cf.* Subsection 3.4, such that,

$$\vartheta^* = \arg\max_\theta\left[p_\theta(\mathbf{x})\right] = \arg\max_\theta\left[g_\theta(\mathbf{x})\right] \tag{25}$$

5. This is referred to as *classification calibrated* by Tewari and Bartlett (2007) and *infinite-sample consistent* by Zhang (2004b).

Tewari and Bartlett (2007) illustrate the two class case of this with reference to equation (24), in that they plot $\ell(A, f(\mathbf{x}))$ against $\ell(B, f(\mathbf{x}))$ where $y(A) = -1$ and $y(B) = +1$. Bartlett, Jordan, and McAuliffe (2004) have shown that provided that this plot is differentiable at $f(\mathbf{x}) = 0$ then consistency is attained. If this is not the case then there is more than one tangent to the plot at $f(\mathbf{x}) = 0$. This is noteworthy because for a particular $\mathbf{x}$, the value of $f(\mathbf{x})$ which minimises the inner product of equation (24) is determined by the point at which a line with slope $-\frac{p_A(\mathbf{x})}{p_B(\mathbf{x})}$ is tangent. If the plot is not differentiable then sample $\mathbf{x}$ with $p_A(\mathbf{x}) > p_B(\mathbf{x})$, and sample $\mathbf{x}'$ with $p_A(\mathbf{x}') < p_B(\mathbf{x}')$ may both have the same value $f(\mathbf{x}) = f(\mathbf{x}') = 0$ even though f minimises the ℓ-risk.

Whereas such a plot is a straightforward 2D plot, in considering a three class case we must turn to 3D surfaces and consider tangent planes with normal given by $\mathbf{p}(\mathbf{x})$ *e.g.* Tewari and Bartlett (2007, Figs.2&3). In these it is illustrated that most 'all-together' methods are inconsistent including those by Weston and Watkins (1999), Weston (1999) and by Crammer and Singer (2001a). However, it is also the case that the approach of Lee et al. (2001, 2004) is consistent.

In order to better understand what is happening, consider the similarities between the method introduced by Weston and Watkins (1999), Weston (1999), and that of Lee et al. (2001, 2004) (§3.4). They both have the same additive approach to forming the loss function (in contrast to the maximum approach of Crammer & Singer, 2001a). The key difference is in their choice of vectors $\mathbf{v}$. While Weston and Watkins (1999), Weston (1999) use, for example $\mathbf{v}_A(B) \propto [\mathbf{y}(B) - \mathbf{y}(A)]$, Lee et al. (2001, 2004) use *e.g.* $\mathbf{v}_A(B) = -\mathbf{y}(A)$. In fact we can also consider using other such vectors — either some combination of these, or more extreme versions. Examples of resulting loss functions are shown in Figure 5, *cf.* Figure 3.

What becomes interesting in terms of Fisher consistency is what happens when these plots of loss contours for all classes are overlaid. Before doing this consider the plots for class C in Figure 5. In particular consider the Lee et al. (2001, 2004) (LLW) case. Here we can clearly identify four regions — that with zero loss, that with loss due only to class A, that with loss due only to class B and that with combined loss. In overlaying contour plots we will similarly seek to identify regions.

This is presented in Figure 6, with the regions seperated by solid black lines. It can be seen that the regions identified in the Weston and Watkins (1999), Weston (1999) and Lee et al. (2001, 2004) plots correspond to planes (and edges) in Figures 2(b) and 3(b) of Tewari and Bartlett (2007). Further, in keeping with their discussion it becomes clear that potential inconsistency problems occur when such region boundaries intersect on the class boundaries. The reason that the Lee et al. (2001, 2004) case manages to avoid this is that region boundaries coincide in this particular setting. The scenarios illustrated in Figure 6 correspond to the following example vectors $\mathbf{v}$, from left to right, top to bottom;

$$\mathbf{v}_A(B) \propto \begin{cases} -0.2\mathbf{y}(B) - \mathbf{y}(A) & \text{The 'Excess LLW' case.} \\ -\mathbf{y}(A) & \text{The 'LLW' case.} \\ \mathbf{y}(B) - 5\mathbf{y}(A) & \text{The 'LLW to WW A' case.} \\ \mathbf{y}(B) - 2\mathbf{y}(A) & \text{The 'LLW to WW B' case.} \\ \mathbf{y}(B) - \mathbf{y}(A) & \text{The 'WW' case.} \\ \mathbf{y}(B) - 0.5\mathbf{y}(A) & \text{The 'Excess WW' case.} \end{cases} \tag{26}$$

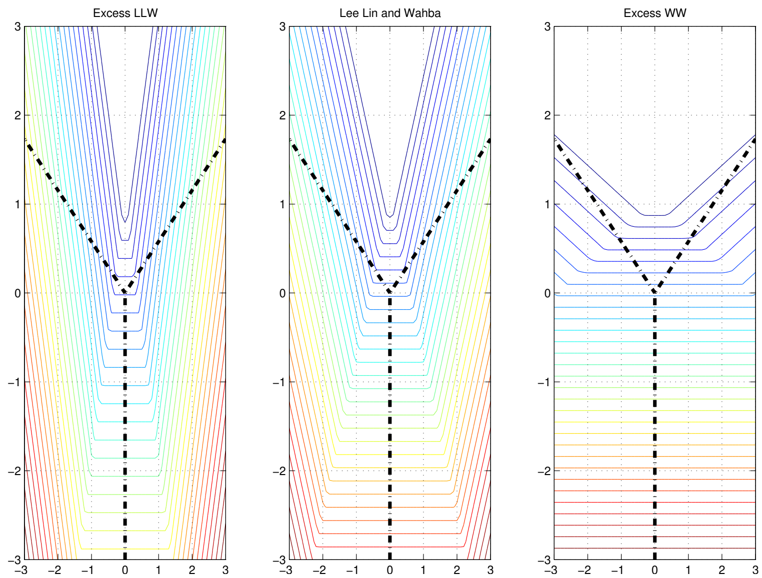


Figure 5: A further illustration of losses with respect to class C. *These correspond to changing vectors* $\mathbf{v}$. *Recall that the Weston and Watkins (1999), Weston (1999) (WW) case had loss contours parallel to class boundaries (Figure 3)*

While it is clear that all cases from 'LLW to WW A' onwards in Figure 6 will be inconsistent, a question remains over the 'Excess LLW' case. To further investigate this we have created plots similar to those in Figures 2 and 3 of Tewari and Bartlett (2007), as shown in Figure 7. From this it is clear from the reverse view that the labelled 'Point of Interest' is again going to pose a consistency problem.

These results give quick geometric insight into why it is that the Lee et al. (2001, 2004) approach appears to be the only Fisher consistent approach involving summed linear losses. We have not performed a similar investigation of the effect of changing $\mathbf{v}$ within the context of the Crammer and Singer (2001a) framework as it seems clear that there will always be a problem at the central point for all reasonable choice of vectors $\mathbf{v}$, *cf.* Tewari and Bartlett (2007, Fig. 2a).

5. Generalisation Bounds

An important aspect of many kernel based algorithms such as SVC, is that Structural Risk Minimisation (SRM) ideas can be applied in order to obtain distribution-free bounds on performance. Such an approach underlies the initial work on SVC in particular, and results in ideas such as the Vapnik Chervonenkis (VC) dimension.

In this section we build on the body of work which is concerned with bounding the performance of multicategory classifiers. This was originally published by Guermeur (2002), but it is important to realise that this paper draws heavily from the work of Elisseeff, Guermeur, and Paugam-Moisy (1999). Further insight is also to be found in work by Paugam-Moisy, Elisseeff, and Guermeur (2000).

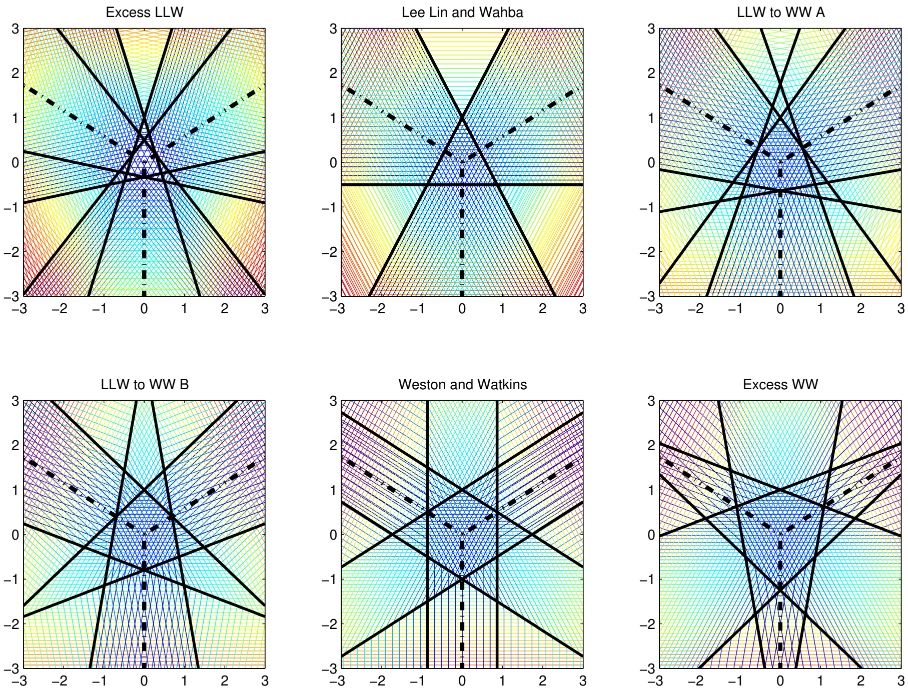
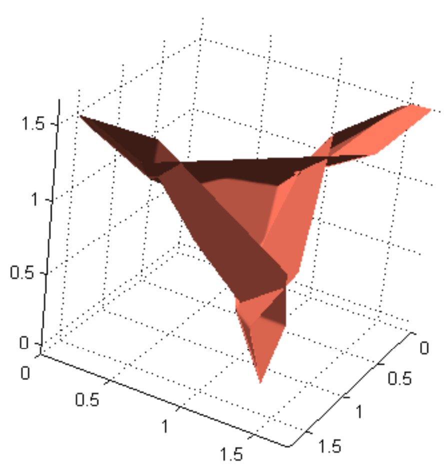
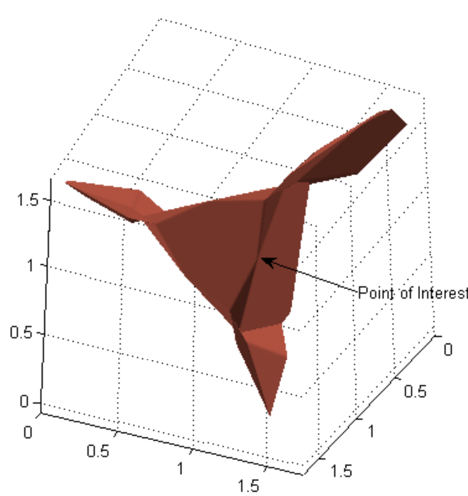


Figure 6: Region Identification — These six different cases represent a rotation of the vectors $\mathbf{v}$, starting at the top left the progression passes through the Lee et al. (2001, 2004) case (middle top), then to the bottom left and through the Weston and Watkins (1999), Weston (1999) case (middle bottom).



(a) Front View



(b) Reverse View

Figure 7: Loss surfaces of the 'Excess LLW' case in Figure 6

By using the geometric approach presented above, it becomes possible to reduce the multidimensional bounding problem to a scalar problem, and thus to fully utilise the more traditional approaches to bounding. These approaches are also drawn on by Elisseeff et al. (1999), however by viewing the problem in the manner proposed here it becomes possible to adopt them virtually unchanged. The key references for this work are by Bartlett (1998) and Williamson, Smola, and Schölkopf (2001).

The final result of this working is to demonstrate that a the bound derived is dependent on the term $\sum_{i=1}^{M-1} \|\mathbf{w}_i\|_F^2$, *cf.* equation (10). This is in keeping with the results of Guermeur (2002), and Elisseeff et al. (1999), as well as traditional two-class bound analyses (Schölkopf & Smola, 2002). Note that some of the notation in this Section is inconsistent with that used elsewhere in this paper, however the difference should be apparent.

5.1 Basic Definitions

We have as a starting reference the canonical function of Elisseeff et al. (1999), Paugam-Moisy et al. (2000), and Guermeur (2002), rewritten in the present notation. In doing this we introduce also the $M-$dimensional vector $\mathbf{y}^c$ which has elements,

$$y_\theta^c = \begin{cases} -1 & \text{if the input has class other than } \theta \\ +1 & \text{if the input has class } \theta. \end{cases}$$

We further introduce the function $\mathbf{g} : \mathcal{X} \to \mathbb{R}^M$,

$$\mathbf{g}(\cdot) = \mathbf{Y}^T \mathbf{f}(\cdot) \tag{27}$$

where $\mathbf{Y}$ is a matrix with columns of class target vectors $\mathbf{y}$, *cf.* Section 3.

Definition 1 (The Original Canonical Function) *Define $R_1(\mathbf{x})$ to be an index such that $g_{R_1(\mathbf{x})}(\mathbf{x}) = \max_\theta g_\theta(\mathbf{x})$ and $R_2(\mathbf{x})$ to be an index such that $g_{R_2(\mathbf{x})}(\mathbf{x}) = \max_{\theta \neq R_1(\mathbf{x})} g_\theta(\mathbf{x})$. The canonical function $\Delta\mathbf{g} : \mathcal{X} \to \mathbb{R}^M$, is then given by,*

$$\Delta g_\theta(\mathbf{x}) = \begin{cases} \frac{1}{2}\left[g_\theta - g_{R_2(\mathbf{x})}\right] = \frac{1}{2}\left[\mathbf{y}(\theta) - \mathbf{y}(R_2(\mathbf{x}))\right]^T \mathbf{f}(\mathbf{x}) = \kappa \mathbf{v}_{R_2(\mathbf{x})}^T(\theta)\mathbf{f}(\mathbf{x}) & \text{if } \theta = R_1(\mathbf{x}) \\ \frac{1}{2}\left[g_\theta - g_{R_1(\mathbf{x})}\right] = \frac{1}{2}\left[\mathbf{y}(\theta) - \mathbf{y}(R_1(\mathbf{x}))\right]^T \mathbf{f}(\mathbf{x}) = \kappa \mathbf{v}_{R_1(\mathbf{x})}^T(\theta)\mathbf{f}(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{28}$$

where κ is a constant of proportionality.

Clearly, if this example has been classified correctly, all the terms $\kappa \mathbf{v}_{R_1(\mathbf{x})}^T(\theta)\mathbf{f}(\mathbf{x})$ should be negative and $\kappa \mathbf{v}_{R_2(\mathbf{x})}^T(R_1(\mathbf{x}))\mathbf{f}(\mathbf{x})$ should be positive. Paugam-Moisy et al. (2000), Guermeur (2002) define the margin by $\varepsilon = \min_\theta y_\theta^c \Delta g_\theta(\mathbf{x})$, however, recall that $\mathbf{v}_A(B) = -\mathbf{v}_B(A)$, and so $\Delta g_{R_1(\mathbf{x})}(\mathbf{x}) = -\Delta g_{R_2(\mathbf{x})}(\mathbf{x})$. Hence, if $R_1(\mathbf{x})$ or $R_2(\mathbf{x})$ is the actual class of $\mathbf{x}$ then $y_{R_1(\mathbf{x})}^c \Delta g_{R_1(\mathbf{x})}(\mathbf{x}) = y_{R_2(\mathbf{x})}^c \Delta g_{R_2(\mathbf{x})}(\mathbf{x})$. In the case that neither of them are the correct class then it can be demonstrated that the margin is not going to be determined by $\Delta g_{R_1(\mathbf{x})}(\mathbf{x})$ uniquely, and so this term does not need to be ever considered. This being the case then the margin is simply given by $\min_{\theta \neq R_1(\mathbf{x})} \kappa \mathbf{v}_\theta^T(R_1(\mathbf{x}))\mathbf{f}(\mathbf{x})$.

This definition of Paugam-Moisy et al. (2000), Guermeur (2002) is somewhat non-intuitive, as it does not make reference to the actual class of the point, merely to $R_1(\mathbf{x})$, which may not be equal to the class ϑ. This is equivalent to defining the margin of the two-class classifier as the absolute value of the function $f(\mathbf{x})$. This is not something which appears in the mainstream texts of Vapnik (1998, p. 402), Schölkopf and Smola (2002, p. 142) or Hastie, Tibshirani, and Friedman (2001, p. 110), for instance. However the general ideas in these and other texts, and the approach in Section 2 can be related to in Definition 2, which is from Paugam-Moisy et al. (2000, Defn. 6) and Guermeur (2002, Defn. 5). In anticipation of this we introduce an alternative canonical function $\Delta\mathbf{f}(\mathbf{x}, \vartheta)$,

$$\Delta\mathbf{f}(\mathbf{x}, \vartheta) = \kappa \cdot \mathbf{V}^T(\vartheta)\mathbf{f}(\mathbf{x}) \tag{29}$$

where $\mathbf{V}(\vartheta)$ is as given in equation (12). If $\mathbf{x}$ is correctly classified then all elements of $\Delta\mathbf{f}$ should be positive.

Definition 2 (Empirical Margin Risk) *Conceptually, the empirical margin risk is the fraction of training samples whose positioning in the $(M-1)-$ dimensional space lies outside their region of zero-loss. Formally this is expressed for some fixed margin $\varepsilon > 0$ and some training set $S = \{\mathbf{x}_i, \vartheta_i\}_{i=1}^N$ as,*

$$R_S^{\varepsilon}(\mathbf{f}) = \frac{1}{N} \left|\{(\mathbf{x}_i, \vartheta_i) \;\; : \;\exists \theta \in (\Theta - \vartheta_i)\,, \; \Delta f_\theta(\mathbf{x}_i, \vartheta_i) < \varepsilon\}\right| \tag{30}$$

A further definition which will be used is that of a pseudo-metric. In this definition note that ℓ_p denotes the norm $\|\mathbf{x}\|_{\ell_p} = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$.

Definition 3 (Pseudo-Metric) *Let $\mathscr{F} : \mathcal{X} \times \Theta \to \mathbb{R}^{M-1}$ be a set of functions and $(\mathbf{f}, \overline{\mathbf{f}}) \in \mathscr{F}^2$. For a set S of points in $\mathcal{X} \times \Theta$, define the pseudo-metric $d_{\ell_\infty,\ell_1}^{\mathscr{F},S}$ by,*

$$d_{\ell_\infty,\ell_1}^{\mathscr{F},S}(\mathbf{f}, \overline{\mathbf{f}}) = \max_{(\mathbf{x},\vartheta)\in S} \sum_{\theta\in(\Theta-\vartheta)} \left|f_\theta(\mathbf{x}, \vartheta) - \overline{f}_\theta(\mathbf{x}, \vartheta)\right|. \tag{31}$$

We now define the covering number (Vapnik, 1998; Schölkopf & Smola, 2002, for example).

Definition 4 (Covering Number) *Let $\left(\mathscr{F}, d^{\mathscr{F},S}\right)$ be a pseudo-metric space, and $B(\hat{\mathbf{f}}, r)$ the closed ball in $\mathscr{F}$ with radius r and centre $\hat{\mathbf{f}}$. The covering number $\mathcal{N}\left(\varepsilon, \mathcal{F}, d^{\mathscr{F},S}\right)$ of a set $\mathcal{F} \in \mathscr{F}$ is the smallest cardinality of set $\overline{\mathcal{F}}$ such that,*

$$\mathcal{F} \subset \bigcup_{\hat{\mathbf{f}}\in\overline{\mathcal{F}}} B(\hat{\mathbf{f}}, \varepsilon). \tag{32}$$

The sets $\overline{\mathcal{F}}$ satisfying this property are called $\varepsilon-$covers of $\mathcal{F}$: each element in $\mathcal{F}$ is at a distance less than ε of an element in $\overline{\mathcal{F}}$. With $|S| = 2N$ then define also,

$$\mathcal{N}_{p,q}\left(\frac{\varepsilon}{2}, \mathcal{F}, 2N\right) = \sup_{S\in\mathcal{X}^{2N}} \mathcal{N}\left(\frac{\varepsilon}{2}, \mathcal{F}, d_{\ell_p,\ell_q}^{\mathscr{F},S}\right).$$

5.2 Presentation of Bounds

We will use this final definition in consideration of,

$$\Delta f^{\varepsilon}(\mathbf{x},\vartheta)=\begin{cases}\varepsilon\cdot\operatorname{sign}\left[\min_{\theta}\Delta f_{\theta}(\mathbf{x},\vartheta)\right], & \text{if } |\min_{\theta}\Delta f_{\theta}(\mathbf{x},\vartheta)|\geq\varepsilon\\ \min_{\theta}\Delta f_{\theta}(\mathbf{x},\vartheta), & \text{otherwise,}\end{cases}\tag{33}$$

which is analogous to the definition of $\Delta\mathbf{g}^{\varepsilon}$ used by Elisseeff et al. (1999, §4), Paugam-Moisy et al. (2000, §6) and Guermeur (2002, §2). This can be used to define the set of scalar-valued functions,

$$\Delta\mathcal{F}^{\varepsilon}=\{\Delta f^{\varepsilon}:\mathbf{f}\in\mathcal{F}\}\tag{34}$$

leading to the Theorem 1, below. The advantage of the approach presented here is that the function Δf^{ε} is a scalar and, as such, it is a lot more straightforward to use the proof structure outlined by Bartlett (1998, Lemma 4) *cf.* Elisseeff et al. (1999, Cor. 2), Paugam-Moisy et al. (2000, Cor. 1), Guermeur (2002, Thm. 1). This is elaborated on, and the two different approaches contrasted by Hill (2007, §4.2,§A.2)

Theorem 1 *With probability at least* $(1-\delta)$*, for every value of* ε *in* $(0,1]$*, the risk* $R(\mathbf{f})$ *of a function* $\mathbf{f}$ *computed by a numerical* $M-$*class discriminant model* $\mathcal{F}$ *trained on a set of size* N *(denoted* S_N*), is bounded above by*

$$R(\mathbf{f})\leq R^{\varepsilon}_{S_N}(\mathbf{f})+\sqrt{\frac{1}{2N}\left[\log\left(2\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},2N\right)\right)+\log\left(\frac{2}{\varepsilon\delta}\right)\right]}+\frac{1}{N}\tag{35}$$

Proof of Theorem 1
The starting point of the proof is equivalent to that in Elisseeff et al. (1999, Eqn. (5)), namely, for any λ,

$$P_{S_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left[R(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right]\geq\lambda\right)\leq 2\times P_{S_N,\widetilde{S}_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R^{\varepsilon}_{\widetilde{S}_N}(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda-\frac{1}{N}\right).\tag{36}$$

Now the aim is to bound the right-hand side of this and the starting point is to consider all permutations σ over $(\mathcal{X}\times\Theta)^{2N}$ such that σ realises a transposition between two elements of the same ranking in $\widetilde{S}_N$ and S_N. Let $\mathcal{U}$ be the uniform distribution over the set of all such permutations σ and so,

$$P_{S_N,\widetilde{S}_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R^{\varepsilon}_{\widetilde{S}_N}(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda-\frac{1}{N}\right)\leq\sup_{S_N,\widetilde{S}_N}\mathcal{U}\left\{\sigma:\sup_{\mathbf{f}\in\mathcal{F}}\left(R^{\varepsilon}_{\widetilde{S}^{\sigma}_N}(\mathbf{f})-R^{\varepsilon}_{S^{\sigma}_N}(\mathbf{f})\right)\geq\lambda-\frac{1}{N}\right\}.$$

Denote a $\frac{\varepsilon}{2}-$cover of the set $\Delta\mathcal{F}^{\varepsilon}$ by $\Delta\overline{\mathcal{F}}^{\varepsilon}$, with elements $\Delta\overline{f}^{\varepsilon}$. Through identical reasoning to that of Bartlett (1998, proof of Lemma 4) by defining,

$$A\left(\Delta\overline{f}^{\varepsilon},S^{\sigma}_N\right)\triangleq\frac{1}{N}\left|\left\{i:\left|\Delta\overline{f}^{\varepsilon}\left(\mathbf{x}^{\sigma}_i,\vartheta^{\sigma}_i\right)-\varepsilon\right|\geq\frac{\varepsilon}{2}\right\}\right|$$

then the above inequality leads to,

$$\begin{aligned}
&P_{S_N,\widetilde{S}_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R^{\varepsilon}_{\widetilde{S}_N}(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda-\frac{1}{N}\right)\\
&\qquad\leq\sup_{S_N,\widetilde{S}_N}\mathcal{U}\left\{\sigma:\sup_{\Delta\overline{f}^{\varepsilon}\in\Delta\overline{\mathcal{F}}^{\varepsilon}}\left(A\left(\Delta\overline{f}^{\varepsilon},S^{\sigma}_N\right)-A\left(\Delta\overline{f}^{\varepsilon},\widetilde{S}^{\sigma}_N\right)\right)\geq\lambda-\frac{1}{N}\right\}\\
&\qquad\leq\left|\Delta\overline{\mathcal{F}}^{\varepsilon}\right|\sup_{\Delta\overline{f}^{\varepsilon}\in\Delta\overline{\mathcal{F}}^{\varepsilon}}\mathcal{U}\left\{\sigma:\left(A\left(\Delta\overline{f}^{\varepsilon},S^{\sigma}_N\right)-A\left(\Delta\overline{f}^{\varepsilon},\widetilde{S}^{\sigma}_N\right)\right)\geq\lambda-\frac{1}{N}\right\}.
\end{aligned}$$

Now by definition $\left|\Delta\overline{\mathcal{F}}^{\varepsilon}\right|=\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},2N\right)$ and so it can be seen that this leads to

$$\begin{aligned}
&P_{S_N,\widetilde{S}_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R^{\varepsilon}_{\widetilde{S}_N}(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda-\frac{1}{N}\right)\\
&\qquad\leq\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},2N\right)\times\sup_{(a_i,b_i)}P\left(\frac{1}{N}\sum_i\left(a_i-b_i\right)\beta_i\geq\lambda-\frac{1}{N}\right)
\end{aligned}\tag{37}$$

where $\beta_i\in\{-1,+1\}$, $P(\beta_i=-1)=P(\beta_i=+1)=0.5$ and they are Independent, and Identically Distributed (IID). Meanwhile

$$a_i=\begin{cases}1 & \text{if }\left|\Delta\overline{f}^{\varepsilon}\left(\widetilde{\mathbf{x}}^{\sigma}_i,\widetilde{\vartheta}^{\sigma}_i\right)-\varepsilon\right|\geq\frac{\varepsilon}{2}\\ 0 & \text{otherwise}\end{cases}$$

and

$$b_i=\begin{cases}1 & \text{if }\left|\Delta\overline{f}^{\varepsilon}\left(\mathbf{x}^{\sigma}_i,\vartheta^{\sigma}_i\right)-\varepsilon\right|\geq\frac{\varepsilon}{2}\\ 0 & \text{otherwise.}\end{cases}$$

Now the right-hand term in equation (37) can be bounded using Hoeffding's inequality (Elisseeff et al., 1999, Theorem 5) such that,

$$P_{S_N,\widetilde{S}_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R^{\varepsilon}_{\widetilde{S}_N}(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda-\frac{1}{N}\right)\leq\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},2N\right)\times\exp\left(-2N\left(\lambda-\frac{1}{N}\right)^2\right).\tag{38}$$

This can be rearranged to demonstrate that,

$$\lambda\leq\sqrt{\frac{1}{2N}\left[\log\left(2\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},2N\right)\right)-\log\left(2P_{S_N,\widetilde{S}_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R^{\varepsilon}_{\widetilde{S}_N}(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda-\frac{1}{N}\right)\right)\right]}+\frac{1}{N}$$

and so, from equation (36)

$$\lambda\leq\sqrt{\frac{1}{2N}\left[\log\left(2\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},2N\right)\right)-\log\left(P_{S_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda\right)\right)\right]}+\frac{1}{N}.$$

Now, with probability at least $(1-\delta)$ where $\delta=P_{S_N}\left(\sup_{\mathbf{f}\in\mathcal{F}}\left(R(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\right)\geq\lambda\right)$, it is the case that $R(\mathbf{f})-R^{\varepsilon}_{S_N}(\mathbf{f})\leq\lambda$. Hence, with probability at least $(1-\delta)$

$$R(\mathbf{f})\leq R^{\varepsilon}_{S_N}(\mathbf{f})+\sqrt{\frac{1}{2N}\left[\log\left(2\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},2N\right)\right)-\log\left(\delta\right)\right]}+\frac{1}{N}.\tag{39}$$

which is analogous to the result of Theorem 4 of Elisseeff et al. (1999). Using this together with Proposition 8 of Bartlett (1998) demonstrates that,

$$R(\mathbf{f}) \leq R^{\varepsilon}_{s_N}(\mathbf{f}) + \sqrt{\frac{1}{2N}\left[\log\left(2\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, 2N\right)\right) + \log\left(\frac{2}{\varepsilon\delta}\right)\right]} + \frac{1}{N} \tag{35}$$

This concludes the proof of Theorem 1.

5.2.1 Bounding $\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, 2N\right)$ using Entropy Numbers

While the generalised risk of Theorem 1 can be bounded by an expression involving the covering number, $\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, 2N\right)$, it is not clear how to determine this number exactly, and a standard approach is to bound it. This is done by following the ideas of Williamson et al. (2001). The first step is to define entropy numbers Guermeur (2002, Defns. 7 & 8), Williamson et al. (2001, eqns. 7-10).

Definition 5 (Entropy Numbers and Operator Norm) *Given a pseudo-metric space* $(\mathscr{F}, d^{\mathscr{F},S}_{\ell_\infty,\ell_1})$ *then, the nth entropy number of a set* $\mathcal{F} \subset \mathscr{F}$ *with respect to* $d^{\mathscr{F},S}_{\ell_\infty,\ell_1}$*, is*

$$\varepsilon_n(\mathcal{F}) \triangleq \inf\left\{\varepsilon > 0 : \mathcal{N}\left(\varepsilon, \mathcal{F}, d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\right) \leq n\right\} \tag{40}$$

The entropy number of an operator $T : \mathscr{F} \to \mathscr{M}$ *follows from the introduction of a unit ball in* $\mathscr{F}$*, denoted* $U_{\mathscr{F}}$*. The nth entropy number of* T *is defined as,*

$$\varepsilon_n(T) \triangleq \varepsilon_n\left(T\left(U_{\mathscr{F}}\right)\right) \tag{41}$$

and the operator norm is given by,

$$\|T\| = \sup_{\mathbf{f}\in U_{\mathscr{F}}} \|T(\mathbf{f})\|_{\mathscr{M}}. \tag{42}$$

To understand the entropy number of T *more explicitly, denote* $T\left(U_{\mathscr{F}}\right)$ *by a set* $\mathcal{M} \in \mathscr{M}$*, and assume some metric* $d^{\mathscr{M}}$*. With these then, in keeping with equation* (40)*, the entropy number is given by* $\varepsilon_n(T) \triangleq \inf\left\{\varepsilon > 0 : \mathcal{N}\left(\varepsilon, \mathcal{M}, d^{\mathscr{M}}\right) \leq n\right\}$*.*

Note that from the first part of this definition it is clear to see that should $\varepsilon_n(\mathcal{F})$ be bounded by some $\overline{\varepsilon}$, then $\mathcal{N}\left(\overline{\varepsilon}, \mathcal{F}, d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\right) \leq n$; Guermeur (2002, Thm. 3), Williamson et al. (2001, Prop. 12). Note also that from Definition 4, in order to bound $\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, 2N\right)$, it is sufficient to bound $\mathcal{N}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\right)$, as discussed in the following Theorem.

Theorem 2 (Bound on $\log\left(\mathcal{N}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\right)\right)$**)** *The log of the covering number* $\mathcal{N}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\right)$ *of a set* $\mathcal{F} \in \mathscr{F}$ *can be bounded by a term proportional to* $\sum_{m=1}^{M-1} \|\mathbf{w}_m\|^2_F$*,* i.e.

$$\log\left(\mathcal{N}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\right)\right) \leq r \sum_{m=1}^{M-1} \|\mathbf{w}_m\|^2_F \tag{43}$$

for some $r > 0$*.*

Proof of Theorem 2

The proof begins with the fact, as highlighted by Williamson et al. (2001, Thm. 10) (see also Guermeur, 2002, Thm. 4), that Maurey's Theorem can be used to bound entropy numbers. For this Theorem note that ℓ_p^q is a vector space containing vectors of dimension q and norm $\|\mathbf{f}\|_{\ell_p^q} = (\sum_{i=1}^q |f_i|^p)^{\frac{1}{p}}$. Furthermore $\mathcal{T}(\mathscr{F}, \mathscr{M})$ denotes the set of all bounded operators between the normed spaces $(\mathscr{F}, \|\cdot\|_{\mathscr{F}})$ and $(\mathscr{M}, \|\cdot\|_{\mathscr{M}})$.

Maurey's Theorem considers that $T \in \mathcal{T}(\mathscr{H}, \ell_\infty^q)$ where $\mathscr{H}$ is a Hilbert space. It then states that there exists a constant $c > 0$ such that, for all $n, q \in \mathbb{N}$,

$$\varepsilon_n(T) \leq c\|T\|\sqrt{\frac{\log\left(1 + \frac{q}{\log n+1}\right)}{\log n + 1}}. \tag{44}$$

While Guermeur (2002) must rely on a generalisation of Maurey's Theorem to matrix output spaces, it is only directly applicable to vector output spaces. This is mentioned by Guermeur (2002), but it is claimed there that it is not a problem, as an extension can be derived, although this is not done. In the current formulation of the problem however, this theorem can be used directly to bound the entropy number, and, as stated, it can also be used to bound the covering number of an operator. These steps are as follows,

$$\varepsilon_n(T) \leq \frac{\varepsilon}{2} = c\|T\|\sqrt{\frac{\log\left(1 + \frac{N}{\log n+1}\right)}{\log n + 1}} \tag{45}$$

$$\mathcal{N}\left(\frac{\varepsilon}{2}, T\left(U_{\mathscr{F}}\right), d_{\ell_\infty,\ell_1}^{\mathscr{M},S}\right) \leq n \tag{46}$$

$$\mathcal{N}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^\varepsilon, d_{\ell_\infty,\ell_1}^{\mathscr{F},S}\right) \leq \mathcal{N}\left(\frac{\varepsilon}{2}, T\left(U_{\mathscr{F}}\right), d_{\ell_\infty,\ell_1}^{\mathscr{M},S}\right). \tag{47}$$

It remains to demonstrate the third of these and, in particular, we aim to do this for

$$T(\mathbf{f}) = \left[\begin{array}{cccc} \min_\theta \Delta f_\theta\left(\mathbf{x}_1, \vartheta_1\right) & \min_\theta \Delta f_\theta\left(\mathbf{x}_2, \vartheta_2\right) & \ldots & \min_\theta \Delta f_\theta\left(\mathbf{x}_N, \vartheta_N\right)\end{array}\right]. \tag{48}$$

This mapping $T : \mathscr{F} \to \mathscr{M}$ is to a vector space which has a norm

$$\|\mathbf{a}\|_{\mathscr{M},S} = \max_{1\leq i\leq |S|} |a_i|. \tag{49}$$

For this case Maurey's Theorem is clearly directly applicable, as are equations (45) and (46). The expressions in equations (48) and (49) are far simpler than their counterparts in Guermeur (2002, §2.3) due to the scalar form of Δf^ε in equation (33). For more on the comparison between the two approaches, see Hill (2007, §4.2.1,§A.2.1). In proving equation (47), consider first,

$$d_{\ell_\infty,\ell_1}^{\mathscr{M},S}\left(T(\mathbf{f}), T(\overline{\mathbf{f}})\right) = \max_{(\mathbf{x},\vartheta)\in S}\left|\min_\theta \Delta f_\theta(\mathbf{x},\vartheta) - \min_\theta \Delta\overline{f}_\theta(\mathbf{x},\vartheta)\right|$$

and, meanwhile,

$$d_{\ell_\infty,\ell_1}^{\mathscr{F},S}\left(\Delta f^\varepsilon, \Delta\overline{f}^\varepsilon\right) = \max_{(\mathbf{x},\vartheta)\in S}\left|\Delta f^\varepsilon(\mathbf{x},\vartheta) - \Delta\overline{f}^\varepsilon(\mathbf{x},\vartheta)\right|$$

and it is clear to see that,

$$\begin{aligned}\max_{(\mathbf{x},\vartheta)\in S}\left|\Delta f^{\varepsilon}(\mathbf{x},\vartheta)-\Delta\overline{f}^{\varepsilon}(\mathbf{x},\vartheta)\right| &\leq \max_{(\mathbf{x},\vartheta)\in S}\left|\min_{\theta}\Delta f_{\theta}(\mathbf{x},\vartheta)-\min_{\theta}\Delta\overline{f}_{\theta}(\mathbf{x},\vartheta)\right| \\ d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\left(\Delta f^{\varepsilon},\Delta\overline{f}^{\varepsilon}\right) &\leq d^{\mathscr{M},S}_{\ell_\infty,\ell_1}\left(T(\mathbf{f}),T(\overline{\mathbf{f}})\right)\end{aligned} \tag{50}$$

which means that, provided $\mathbf{f},\overline{\mathbf{f}}\in U_{\mathscr{F}}$, then equation (47) is correct. An extended version of this derivation is presented by Hill (2007, §B.2).

All that remains is to bound $\|T\|$. Now, from equation (49),

$$\|T(\mathbf{f})\|_{\mathscr{M},S}=\max_{i\leq i\leq|S|}\left(\min_{\theta\neq\vartheta_i}\left|\mathbf{v}_{\theta}^{T}(\vartheta_i)\mathbf{W}\Phi(\mathbf{x}_i)\right|\right). \tag{51}$$

Through the Cauchy-Schwarz inequality (Guermeur, 2002, §A.2) and with $\Lambda_{\mathcal{X}}$ being the radius of a ball including $\Phi(\mathcal{X})$, then

$$\|T(\mathbf{f})\|_{\mathscr{M},S}\leq\Lambda_{\mathcal{X}}\max_{\vartheta}\left(\min_{\theta\neq\vartheta}\|\mathbf{v}_{\theta}^{T}(\vartheta)\mathbf{W}\|_2\right) \tag{52}$$

this means that,

$$\begin{aligned}\|T\| &\leq \Lambda_{\mathcal{X}}\max_{\vartheta}\left(\min_{\theta\neq\vartheta}\|\mathbf{v}_{\theta}^{T}(\vartheta)\mathbf{W}\|_2\right) \\ &\leq \Lambda_{\mathcal{X}}\sqrt{\sum_{m=1}^{M-1}\|\mathbf{w}_m\|_F^2}.\end{aligned} \tag{53}$$

In this we have made the assumption that $\mathbf{f}\in U_{\mathscr{F}}$, however if this is not the case it is straightforward to arrive at an analogous solution. More on this can be found in Williamson et al. (2001, §V.B), Elisseeff et al. (1999, Prop. 4) or Guermeur (2002, Prop. 1), for example. Combining this result with equation (45)

$$\varepsilon_n(T_F)\leq\frac{\varepsilon}{2}\leq c\Lambda_{\mathcal{X}}\sqrt{\sum_{m=1}^{M-1}\|\mathbf{w}_m\|_F^2}\sqrt{\frac{\log\left(1+\frac{N}{\log n+1}\right)}{\log n+1}}$$

which can be rearranged to give,

$$\log n\leq\frac{4c^2\Lambda_{\mathcal{X}}^2\log(1+N)\sum_{m=1}^{M-1}\|\mathbf{w}_m\|_F^2}{\varepsilon^2}-1$$

when $n\geq 1$, which is always going to be the case as this is a bound on a covering number. Equation (46) then gives,

$$\log\left(\mathcal{N}\left(\frac{\varepsilon}{2},T_F\left(U_{\mathscr{F}}\right),d^{\mathscr{M}_F,S}_{\ell_\infty,\ell_1}\right)\right)\leq\frac{4c^2\Lambda_{\mathcal{X}}^2\log(1+N)\sum_{m=1}^{M-1}\|\mathbf{w}_m\|_F^2}{\varepsilon^2}-1$$

and so, finally, from equation (47)

$$\log\left(\mathcal{N}\left(\frac{\varepsilon}{2},\Delta\mathcal{F}^{\varepsilon},d^{\mathscr{F},S}_{\ell_\infty,\ell_1}\right)\right)\leq\frac{4c^2\Lambda_{\mathcal{X}}^2\log(1+N)\sum_{m=1}^{M-1}\|\mathbf{w}_m\|_F^2}{\varepsilon^2}-1.$$

This demonstrates the result in equation (43) of Theorem 2 and moreover shows that the constant r is given by,

$$r = \frac{4c^2 \Lambda_{\mathcal{X}}^2 \log(1+N)}{\varepsilon^2}. \tag{54}$$

This concludes the proof of Theorem 2

5.3 Summary of Generalisation Bounds

In Subsection 5.2, Theorem 1 showed that the risk $R(\mathbf{f})$ of a function $\mathbf{f}$ is bounded with probability at least $(1-\delta)$ by,

$$R(\mathbf{f}) \leq R_{S_N}^{\varepsilon}(\mathbf{f}) + \sqrt{\frac{1}{2N}\left[\log\left(2\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, 2N\right)\right) + \log\left(\frac{2}{\varepsilon\delta}\right)\right]} + \frac{1}{N}. \tag{35}$$

Where, from Definition 4; $\mathcal{N}_{\infty,1}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, 2N\right) = \sup_{S\in\mathcal{X}^{2N}} \mathcal{N}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, d_{\ell_\infty,\ell_1}^{\mathscr{F},S}\right)$, and from Theorem 2

$$\log\left(\mathcal{N}\left(\frac{\varepsilon}{2}, \Delta\mathcal{F}^{\varepsilon}, d_{\ell_\infty,\ell_1}^{\mathscr{F},S}\right)\right) \leq r\sum_{m=1}^{M-1} \|\mathbf{w}_m\|_F^2 \tag{43}$$

where r is positive and given by equation (54). As a result

$$R(\mathbf{f}) \leq R_{S_N}^{\varepsilon}(\mathbf{f}) + \sqrt{\frac{1}{2N}\left[r\sum_{m=1}^{M-1} \|\mathbf{w}_m\|_F^2 + \log\left(\frac{4}{\varepsilon\delta}\right)\right]} + \frac{1}{N}. \tag{55}$$

As mentioned in the derivation of this result, the methodology employed has been more in keeping with the two-class derivation than the bound derived by Guermeur (2002). This is due to the use of the scalar function Δf^{ε}, as introduced in equation (33). The use of this function is a logical consequence of viewing the problem in the geometric framework of Section 2. It allows T to be a mapping to a vector space, as it is in the two-class case, rather than to a matrix space, as it is in the work by Guermeur (2002).

Not only does the use of Δf^{ε} simplify the working, but the final result is that the derived bound is tighter than that of Guermeur (2002). This has been rederived in the present notation by Hill (2007, App. A) in which the assumption that Maurey's Theorem is applicable directly is maintained. In presenting it here we first define $\overline{\Theta}$ to be the set of $\frac{M(M-1)}{2}$ unique class combinations. If the class pair $(\phi,\varphi) \in \overline{\Theta}$ then $(\varphi,\phi) \notin \overline{\Theta}$ and the equivalent expression to equation (55) is,

$$R(\mathbf{f}) \leq R_{S_N}^{\varepsilon}(\mathbf{f}) + \sqrt{\frac{1}{2N}\left[r\sum_{(\phi,\varphi)\in\overline{\Theta}} \|\mathbf{v}_{\phi}^T(\varphi)\mathbf{W}\|^2 + \log\left(\frac{4}{\varepsilon\delta}\right)\right]} + \frac{1}{N}. \tag{56}$$

where now

$$r = \frac{8c^2\kappa^2\Lambda_{\mathcal{X}}^2 M(M-1)\log(1+N)}{\varepsilon^2}.$$

6. Other Kernel-Based Methods

In this section the use of the framework presented in Section 2 is described with respect to ν-SVC, LS-SVC, LSVC, PSVC, and BPM.

6.1 ν-Support Vector Classification

In this case the two-class optimisation problem (Schölkopf & Smola, 2002) is to

$$\text{Minimise} \left(\frac{1}{2}\|\mathbf{w}\|_F^2 + \sum_{i=1}^{N} \xi_i - \nu\varepsilon\right), \text{ Subject to } \begin{cases} y_i \left[\langle \Phi(\mathbf{x}_i), \mathbf{w}\rangle_F + b\right] \geq \varepsilon - \xi_i \\ \xi_i \geq 0, \text{ and, } \varepsilon \geq 0 \end{cases} \tag{57}$$

and the extension to the polychotomous case is straightforward, namely to

$$\begin{aligned} &\text{Minimise} \left(\frac{1}{2}\sum_{m=1}^{M-1} \|\mathbf{w}_m\|_F^2 + \sum_{i=1}^{N} \sum_{\theta\in(\Theta-\vartheta_i)} \xi_{i,\theta} - \sum_{\vartheta\in\Theta} \sum_{\theta\in(\Theta-\vartheta)} \nu\varepsilon_\theta(\vartheta)\right) \\ &\text{Subject to } \begin{cases} \sum_{m=1}^{M-1} v_{\theta,m}(\vartheta_i) \left[\langle \Phi(\mathbf{x}_i), \mathbf{w}_m\rangle_F + b_m\right] \geq \varepsilon - \xi_{i,\theta} \\ \xi_{i,\theta} \geq 0, \text{ and, } \varepsilon \geq 0. \end{cases} \end{aligned} \tag{58}$$

Following the usual Lagrangian dual approach results in the final aim being to maximise

$$L_D = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{m=1}^{M-1} \boldsymbol{\alpha}_i^T \mathbf{V}^T(\vartheta_i)\mathbf{V}(\vartheta_j)\boldsymbol{\alpha}_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{59}$$

subject to $0 \leq \alpha_{i,\theta} \leq 1$, $\forall i, \theta \in (\Theta - \vartheta_i)$, $\sum_{i=1}^{N} \mathbf{V}(\vartheta_i)\boldsymbol{\alpha}_i = \mathbf{0}$, and $\sum_{i=1}^{N}\sum_{\theta\in(\Theta-\vartheta_i)} \alpha_{i,\theta} > \nu$. The output is as given in equation (19).

6.2 Least Squares Support Vector Classification

LS-SVC as developed at length by Van Gestel, Suykens, Baesens, Viaene, Vanthienen, Dedene, De Moor, and Vandewalle (2001) is much the same as standard SVC, except that the empirical loss is now taken to be quadratic; see the top-left corner of Figure 3, and equation (6). Multiclass versions have been published (Van Gestel, Suykens, Lanckriet, Lambrechts, De Moor, & Vandewalle, 2002) which rely on coding schemes as discussed in Subsection 3.3. The two-class case aims to

$$\text{Minimise} \left(\frac{1}{2}\|\mathbf{w}\|_F^2 + C\sum_{i=1}^{N} \xi_i^2\right), \text{ Subject to } y_i \left[\langle \Phi(\mathbf{x}_i), \mathbf{w}\rangle_F + b\right] = 1 - \xi_i \tag{60}$$

An alternative multi-category extension to the coding approach exists, *i.e.*

$$\begin{aligned} &\text{Minimise} \left(\frac{1}{2}\sum_{m=1}^{M-1} \|\mathbf{w}_m\|_F^2 + C\sum_{i=1}^{N} \sum_{\theta\in(\Theta-\vartheta_i)} \xi_{i,\theta}^2\right) \\ &\text{Subject to } \sum_{m=1}^{M-1} v_{\theta,m}(\vartheta_i) \left[\langle \Phi(\mathbf{x}_i), \mathbf{w}_m\rangle_F + b_m\right] = \varepsilon_\theta(\vartheta_i) - \xi_{i,\theta}. \end{aligned} \tag{61}$$

Now, define

$$\begin{gathered}
\boldsymbol{\alpha}' = \left[\begin{array}{ccc} \boldsymbol{\alpha}_1^T & \dots & \boldsymbol{\alpha}_N^T \end{array} \right]^T \qquad \boldsymbol{\varepsilon}' = \left[\begin{array}{ccc} \boldsymbol{\varepsilon}^T(\vartheta_1) & \dots & \boldsymbol{\varepsilon}^T(\vartheta_N) \end{array} \right]^T \\
\mathbf{Z}_m = \left[\begin{array}{ccc} \Phi(\mathbf{x}_1)\mathbf{v}_m^{*T}(\vartheta_1) & \dots & \Phi(\mathbf{x}_N)\mathbf{v}_m^{*T}(\vartheta_N) \end{array} \right] \qquad \mathbf{V}' = \left[\begin{array}{ccc} \mathbf{V}(\vartheta_1) & \dots & \mathbf{V}(\vartheta_N) \end{array} \right] \\
\mathbf{Z}' = \left[\begin{array}{ccc} \mathbf{Z}_1^T & \dots & \mathbf{Z}_{M-1}^T \end{array} \right]^T .
\end{gathered}$$

With these definitions then it can be shown (Van Gestel et al., 2001) that the optimisation problem becomes equivalent to finding $\boldsymbol{\alpha}'$ and $\mathbf{b}$ to satisfy,

$$\left[\begin{array}{cc} \mathbf{0} & \mathbf{V}' \\ \mathbf{V}'^T & \mathbf{Z}'^T\mathbf{Z}' + C\mathbf{I} \end{array} \right] \left[\begin{array}{c} \mathbf{b} \\ \boldsymbol{\alpha}' \end{array} \right] = \left[\begin{array}{c} \mathbf{0} \\ \boldsymbol{\varepsilon}' \end{array} \right]. \tag{62}$$

The classifier is found by solving these linear equations. Note that finding $\mathbf{Z}'^T\mathbf{Z}'$ does not require reference to the feature space, but only kernel evaluations. The final output is again as in equation (19).

6.3 Lagrangian Support Vector Classification

As introduced by Mangasarian and Musicant (2001), the LSVC is an algorithm which has its strength in that it is computationally efficient, and easy to implement. It again uses a quadratic empirical loss, as illustrated in the top-left corner of Figure 3, and detailed in equation (6). The method for two-class classification aims to

$$\text{Minimise } \left(\frac{1}{2}\left[\|\mathbf{w}\|_F^2 + b^2 \right] + C\sum_{i=1}^{N} \xi_i^2 \right), \text{ Subject to } y_i \left[\langle \Phi(\mathbf{x}_i), \mathbf{w} \rangle_F + b \right] \geq 1 - \xi_i. \tag{63}$$

This can be reformulated to a multi-category problem resulting in,

$$\begin{gathered}
\text{Minimise } \left(\frac{1}{2}\sum_{m=1}^{M-1} \left[\|\mathbf{w}_m\|_F^2 + b_m^2 \right] + C\sum_{i=1}^{N} \sum_{\theta \in (\Theta - \vartheta_i)} \xi_{i,\theta}^2 \right) \\
\text{Subject to } \sum_{m=1}^{M-1} v_{\theta,m}(\vartheta_i) \left[\langle \Phi(\mathbf{x}_i), \mathbf{w}_m \rangle_F + b_m \right] \geq \varepsilon_\theta(\vartheta_i) - \xi_{i,\theta}.
\end{gathered} \tag{64}$$

The dual to this is,

$$L_D = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{\alpha}_i^T \mathbf{V}^T(\vartheta_i)\mathbf{V}(\vartheta_j)\boldsymbol{\alpha}_j \left[K(\mathbf{x}_i, \mathbf{x}_j) + 1 \right] + \sum_{i=1}^{N} \boldsymbol{\alpha}_i^T \left[\boldsymbol{\varepsilon}(\vartheta_i) - \frac{1}{2C}\boldsymbol{\alpha}_i \right] \tag{65}$$

which needs to be maximised subject to $\alpha_{i,\theta} \geq 0$ for all i, and all $\theta \in (\Theta - \vartheta_i)$. Once that has been done then $\mathbf{b} = \sum_{i=1}^{N} \mathbf{V}(\vartheta_i)\boldsymbol{\alpha}_i$. The final solution again takes the form of equation (19).

6.4 Proximal Support Vector Classification

Following on from the LSVC method, the PSVC approach was developed by Fung and Mangasarian (2001b, 2001a). While they have presented a multi-category approach it is a one-against-all algorithm, not an 'all-together' one. The two-class aim is to

$$\text{Minimise } \left(\frac{1}{2}\left[\|\mathbf{w}\|_F^2 + b^2\right] + C\sum_{i=1}^{N}\xi_i^2\right), \text{ Subject to } y_i\left[\langle\Phi(\mathbf{x}_i),\mathbf{w}\rangle_F + b\right] = 1 - \xi_i. \quad (66)$$

which is, once more, the same as that for LS-SVC in Subsection 6.2 except for the b^2 term. This can be reformulated to a multi-category problem resulting in,

$$\begin{aligned} &\text{Minimise } \left(\frac{1}{2}\sum_{m=1}^{M-1}\left[\|\mathbf{w}_m\|_F^2 + b_m^2\right] + C\sum_{i=1}^{N}\sum_{\theta\in(\Theta-\vartheta_i)}\xi_{i,\theta}^2\right) \\ &\text{Subject to } \sum_{m=1}^{M-1} v_{\theta,m}(\vartheta_i)\left[\langle\Phi(\mathbf{x}_i),\mathbf{w}_m\rangle_F + b_m\right] = \varepsilon_\theta(\vartheta_i) - \xi_{i,\theta}. \end{aligned} \quad (67)$$

Now, define $\mathbf{v}^{*\prime} = \left[\begin{array}{cccccc} \mathbf{v}_1^{*T}(\vartheta_1) & \dots & \mathbf{v}_1^{*T}(\vartheta_N) & \mathbf{v}_2^{*T}(\vartheta_1) & \dots & \mathbf{v}_N^{*T}(\vartheta_N)\end{array}\right]^T$, and with this then as for LS-SVC the optimisation problem has an exact solution,

$$\boldsymbol{\alpha}' = \left(\mathbf{I} + \mathbf{Z}'^T\mathbf{Z}' + \mathbf{v}^{*\prime}\mathbf{v}^{*\prime T}\right)^{-1}\boldsymbol{\varepsilon}' \quad (68)$$

where everything is as defined in Section 6.2 and $\mathbf{b} = \sum_{i=1}^{N}\mathbf{V}(\vartheta_i)\boldsymbol{\alpha}_i$. As before, the final solution takes the form of equation (19).

6.5 Bayes Point Machines

BPMs were introduced by Herbrich, Graepel, and Campbell (2000a) and the ideas can be extended to a multi-category problem. In short they consider what they term *Version Space*, $\mathcal{V}$. In the two-class case this is the region in which a weight vector $\mathbf{w}$ can lie without inducing any classification errors on the training set.

Within version space a uniform distribution is assumed over all possible linear (in feature space) classifiers, h, outside it is assumed zero. The *Bayes point classifier* is then given by

$$h_{bp} = \arg\min_{h\in\mathcal{H}}\mathbb{E}_X\left[\mathbb{E}_{H|\{\mathbf{x}_i,y_i\}_{i=1}^N}\left[\ell\left(h(X),H(X)\right)\right]\right] \quad (69)$$

where $\ell(\cdot,\cdot)$ is some loss function (typically the zero-one loss function is used) and the inner expectation is over classifiers $H\in\mathcal{H}$. One problem with this definition is that it is not usual that there is any knowledge about P_X and so evaluation of $\mathbb{E}_X$ is impossible. With some assumptions about the form of P_X (see Herbrich et al., 2000a, for more) it can, however, be shown that the centre of mass,

$$\mathbf{w}_{cm} = \frac{\mathbb{E}_{\mathbf{w}|\{\mathbf{x}_i,y_i\}_{i=1}^N}[\mathbf{w}]}{\|\mathbb{E}_{\mathbf{w}|\{\mathbf{x}_i,y_i\}_{i=1}^N}[\mathbf{w}]\|} \quad (70)$$

is a good approximation to $\mathbf{w}_{bp}$. Eventually the problem becomes to identify $\mathcal{V}$, which is some contiguous and convex space, and then to find $\mathbf{w}_{cm}$ given that there is a uniform distribution assumed over the weight vectors in this space.

Note that version space is defined by

$$\mathcal{V} = \{\mathbf{w} : y_i\langle\Phi(\mathbf{x}_i), \mathbf{w}\rangle > 0,\ \|\mathbf{w}\| = 1,\ \forall i\}\,, \tag{71}$$

When considering multiple classes then the condition $y_i\langle\Phi(\mathbf{x}_i), \mathbf{w}\rangle > 0$ becomes $\mathbf{V}^T(\vartheta_i)\mathbf{W}\Phi(\mathbf{x}_i) > \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}^T$ where the inequality indicates component-wise inequalities, and the matrix $\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & \dots & \mathbf{w}_{M-1} \end{bmatrix}^T$ has been introduced. As a result the version space is given by

$$\mathcal{V} = \left\{(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{M-1}) : \mathbf{V}^T(\vartheta_i)\mathbf{W}\Phi(\mathbf{x}_i) > \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}^T,\ \|\mathbf{w}_m\| = 1\ \forall m, i\right\}, \tag{72}$$

which is identical in form to equation (71). Extensions of the *kernel billiards* algorithm described by Herbrich et al. (2000a) can be used to find $\mathbf{W}_{cm}$, which is analogous to $\mathbf{w}_{cm}$ in equation (70). Their method for including training errors can also be seamlessly incorporated.

7. Implementation through Sequential Minimal Optimisation

The geometric construction introduced in Section 2 allows insight into the multi-category problem, which should motivate alternative approaches to efficiently solve the 'all-together' optimisation problem. One possibility for SVC is presented here, based on a vector-valued version of SMO which was first introduced for binary classification by Platt (1999). This has several advantages, including the fact that it is reasonably straightforward to understand, relatively easy to implement, quite efficient and flexible, in addition to being well established and known.

SMO optimises with respect to two points at a time, denote these c and d. With this notation, and with $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, then the dual Lagrangian in equation (13) becomes

$$\begin{aligned} L = \left(\boldsymbol{\alpha}_c^T\boldsymbol{\varepsilon}(\vartheta_c) + \boldsymbol{\alpha}_d^T\boldsymbol{\varepsilon}(\vartheta_d)\right) - \frac{1}{2}\boldsymbol{\alpha}_c^T\mathbf{V}^T(\vartheta_c)\mathbf{V}(\vartheta_c)\boldsymbol{\alpha}_c K_{cc} - \frac{1}{2}\boldsymbol{\alpha}_d^T\mathbf{V}^T(\vartheta_d)\mathbf{V}(\vartheta_d)\boldsymbol{\alpha}_d K_{dd} \\ - \boldsymbol{\alpha}_c^T\mathbf{V}^T(\vartheta_c)\mathbf{V}(\vartheta_d)\boldsymbol{\alpha}_d K_{cd} - \boldsymbol{\alpha}_c^T\mathbf{V}^T(\vartheta_c)\mathbf{z}_c - \boldsymbol{\alpha}_d^T\mathbf{V}^T(\vartheta_d)\mathbf{z}_d + \text{constant.} \end{aligned} \tag{73}$$

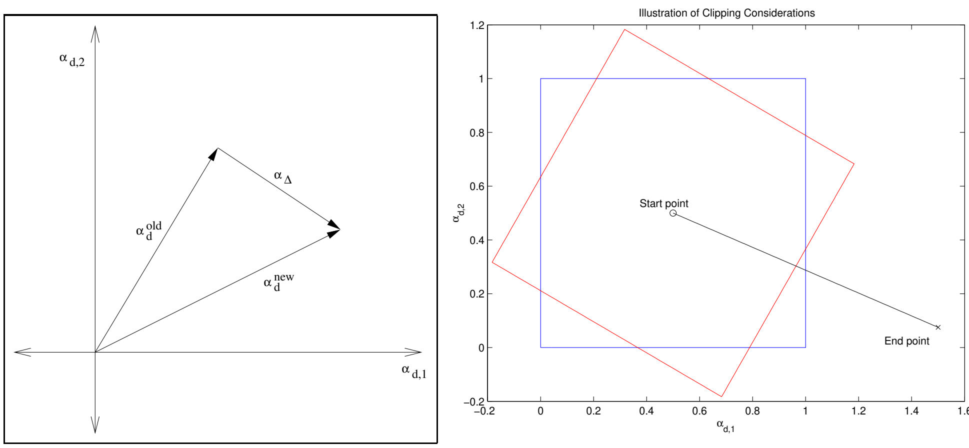
where $\mathbf{z}_c$ is a vector with elements $z_{c,m} = \sum_{i=1, i\neq c,d}^{N} \boldsymbol{\alpha}_i^T\mathbf{v}_m^*(\vartheta_i)K_{ic}$ and similarly for $z_{d,m}$. As shown by Hill and Doucet (2005, §6); by expressing $\boldsymbol{\alpha}_c$ in terms of $\boldsymbol{\alpha}_d$, through the constraint in equation (17), and finding a minimum by setting $\nabla_{\boldsymbol{\alpha}_d}L(\boldsymbol{\alpha}_d) = 0$, then an SMO update takes the form,

$$\boldsymbol{\alpha}_d^{new} = \boldsymbol{\alpha}_d^{old} + \frac{\mathbf{V}^{-1}(\vartheta_d)}{K_{dd} + K_{cc} - 2K_{dc}}\left[\boldsymbol{\psi}(\mathbf{x}_c) - \boldsymbol{\psi}(\mathbf{x}_d) + \mathbf{V}^{-T}(\vartheta_d)\boldsymbol{\varepsilon}(\vartheta_d) - \mathbf{V}^{-T}(\vartheta_c)\boldsymbol{\varepsilon}(\vartheta_c)\right] \tag{74}$$

where, $\mathbf{V}^{-1}(\vartheta_d)$ exists provided that the vectors $\{\mathbf{v}_\theta(\vartheta_d) : \theta \in (\Theta - \vartheta_d)\}$ are linearly independent, which is nearly always the case, although it is possible to conceive of pathological cases. Recall that $\boldsymbol{\psi}$ was introduced in equation (9) *cf.* (19).

7.1 Clipping

Recall that all elements of $\boldsymbol{\alpha}_c$ and $\boldsymbol{\alpha}_d$ are upper and lower bounded (equation (16)), and hence clipping may be required. This is best understood through Figure 8, which relates



(a) A vector illustration of a proposed update

(b) Constraints within which the update must be made

Figure 8: The proposed update for a three-class problem. *The new point is shown in Subfigure 8(b) to be outside the allowed regions. These correspond to the overall limits of* 0 *and* $CD_1(\vartheta_d)$ *(upright box) and limits imposed by* $\boldsymbol{\alpha}_c$ *considerations (tilted box).*

to the three-class case, however the ideas are generically applicable. Given the update in equation (74) of the form $\boldsymbol{\alpha}_d^{new} = \boldsymbol{\alpha}_d^{old} + \boldsymbol{\alpha}_\Delta$, if $\boldsymbol{\alpha}_c^{new}$ or $\boldsymbol{\alpha}_d^{new}$ lie outside their constraints then the line between $\boldsymbol{\alpha}_d^{old}$ and $\boldsymbol{\alpha}_d^{new}$ is traced back along until this is no longer the case. Ultimately some $\kappa \in [0, 1)$ is found such that

$$\boldsymbol{\alpha}_d^{new,clipped} = \boldsymbol{\alpha}_d^{old} + \kappa \boldsymbol{\alpha}_\Delta.$$

As the optimisation surface is convex, improvements are still made with every update.

7.2 Updating Non-Extremal Components

Often, updates of the form of equation (74) involve vectors $\boldsymbol{\alpha}_c^{old}$ and $\boldsymbol{\alpha}_d^{old}$ which have extremal components (*i.e.* components at their constraint-introduced limits). This can lead to a computational bottleneck as any update which suggests that these components should lie further outside the allowed region will result in the clipping procedure returning the original vectors.

To avoid this consider again that the two points to update are labelled c and d, and denote the number of non-extremal components of each as P_c and P_d respectively. An update is likely possible[6] if $P_d > M - 1 - P_c$ and, this being the case, let $P_d + P_c + 1 - M$

6. Otherwise there is only one solution, the current one.

non-extremal components of $\boldsymbol{\alpha}_d$ be grouped into a new vector $\overline{\boldsymbol{\alpha}}_d$. The remaining elements of both $\boldsymbol{\alpha}_c$ and $\boldsymbol{\alpha}_d$ are dependent on these. Owing to the linearity of the relationship between $\boldsymbol{\alpha}_c$ and $\boldsymbol{\alpha}_d$, as introduced by the constraint in equation (17) then it becomes apparent that

$$\boldsymbol{\alpha}_d = \widetilde{\boldsymbol{\alpha}}_d + \mathbf{A}_d\overline{\boldsymbol{\alpha}}_d \quad \text{and} \quad \boldsymbol{\alpha}_c = \widetilde{\boldsymbol{\alpha}}_c + \mathbf{A}_c\overline{\boldsymbol{\alpha}}_d \tag{75}$$

describe dependencies, for some $\widetilde{\boldsymbol{\alpha}}_d$, $\mathbf{A}_d$, $\widetilde{\boldsymbol{\alpha}}_c$, and $\mathbf{A}_c$. Of these $\widetilde{\boldsymbol{\alpha}}_c$ and $\widetilde{\boldsymbol{\alpha}}_d$ contain the extremal components which will not be updated, together with zeros, and $\mathbf{A}_c$ and $\mathbf{A}_d$ are matrices consisting of ones and zeros which map the variable components back to their original positions in the vectors $\boldsymbol{\alpha}_c$ and $\boldsymbol{\alpha}_d$. It can be shown (Hill & Doucet, 2005, App. E), that the SMO update in this case is,

$$\begin{aligned}\overline{\boldsymbol{\alpha}}_d^{new} = \overline{\boldsymbol{\alpha}}_d^{old} + & \frac{(\mathbf{A}_d^T\mathbf{V}^T(\vartheta_d)\mathbf{V}(\vartheta_d)\mathbf{A}_d)^{-1}\mathbf{A}_d^T\mathbf{V}^T(\vartheta_d)}{K_{dd}+K_{cc}-2K_{dc}} \\ & \times \left[\boldsymbol{\psi}(\mathbf{x}_c) - \boldsymbol{\psi}(\mathbf{x}_d) + \mathbf{V}^{-T}(\vartheta_d)\boldsymbol{\varepsilon}(\vartheta_d) - \mathbf{V}^{-T}(\vartheta_c)\boldsymbol{\varepsilon}(\vartheta_c)\right].\end{aligned} \tag{76}$$

Again, clipping can be performed as in Subsection 7.1. Note that in this expression only the evaluations of $\boldsymbol{\psi}(\cdot)$ actually change during the optimisation process. All others, especially the matrix-valued numerator may be held in memory to speed the procedure, where possible.

7.3 Point Selection

It remains to select points c and d. Platt (1999), presents a number of heuristics, however the improvements suggested by Keerthi, Shevade, Bhattacharyya, and Murthy (2001) appear to be more efficient and will form the basis of that overviewed here. In the binary case the essential approach is to identify points requiring the highest and lowest offsets b in order that their underlying function $f(\cdot)$ is as might be expected, *i.e.* it has the sign of the relevant point. When considering two classes, A and B in the multi-category arrangement a directly analogous approach can be taken in that the problem is reduced to a two-class problem across their mutual boundary, and a comparable scalar metric can be found.

The starting point in this methodology is to construct the Lagrangian which governs the dual optimisation problem as given in equation (13);

$$\begin{aligned}L =& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\boldsymbol{\alpha}_i^T\mathbf{V}^T(\vartheta_i)\mathbf{V}(\vartheta_j)\boldsymbol{\alpha}_j^T K(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\boldsymbol{\alpha}_i^T\boldsymbol{\varepsilon}(\vartheta_i) - \sum_{i=1}^{N}\boldsymbol{\alpha}_i^T\boldsymbol{\delta}_i \\ & + \sum_{i=1}^{N}\sum_{\theta\in(\Theta-\vartheta_i)}\mu_{i,\theta}\left(\alpha_{i,\theta} - CD_\theta(\vartheta_i)\right) - \sum_{i=1}^{N}\boldsymbol{\alpha}_i^T\mathbf{V}^T(\vartheta_i)\boldsymbol{\eta}_m\end{aligned}$$

where $\{\delta_{i,\theta}, \mu_{i,\theta} : i \in \{1,\ldots,N\}, \theta \in (\Theta - \vartheta_i)\}$ and $\{\eta_m : m \in \{1,\ldots,(M-1)\}\}$ are Lagrangian multipliers. Differentiating this with respect to $\boldsymbol{\alpha}_i$ and setting the result equal to zero implies that,

$$\mathbf{V}^T(\vartheta_i)\left(\boldsymbol{\psi}(\mathbf{x}_i) + \boldsymbol{\eta}\right) = \boldsymbol{\varepsilon}(\vartheta_i) + \boldsymbol{\delta}_i - \boldsymbol{\mu}_i \tag{77}$$

and, hence,

$$\boldsymbol{\eta} = \mathbf{V}^{-T}(\vartheta_i)\left(\boldsymbol{\varepsilon}(\vartheta_i) + \boldsymbol{\delta}_i - \boldsymbol{\mu}_i\right) - \boldsymbol{\psi}(\mathbf{x}_i). \tag{78}$$

```
while KKT conditions not satisfied.

   for all combinations of two classes (denoted A and B).

      Perform two-class SMO along the direction v_B(A) with
      updates given by equation (76) and i_up, i_low found through (79)
```

Table 1: Algorithm Pseudo-Code

Consider the update of two points of classes A and B and let $\vartheta_i = A$, in doing this recall the equality, as discussed by Keerthi et al. (2001), of $\boldsymbol{\eta}$ and $\mathbf{b}$. With this in mind it becomes apparent that an equivalent metric for updating is the difference between the respective $\boldsymbol{\eta}$ values across the boundary between the two classes. To find the perpendicular distance, take the inner product with the perpendicular to the boundary, for instance $\mathbf{v}_B(A)$

$$\mathbf{v}_B^T(A)\boldsymbol{\eta} = \varepsilon_B(A) + \delta_{i,B} - \mu_{i,B} - \mathbf{v}_B^T(A)\boldsymbol{\psi}(\mathbf{x}_i). \tag{79}$$

This expression is now directly comparable to the equivalent key starting point in the point selection process as it is directly analogous to the scalar used by Keerthi et al. (2001). Indeed the parameters b_{up} and b_{low} used there are equivalent to the extreme values of $\mathbf{v}_B^T(A)\boldsymbol{\eta}$.

7.4 Multi-Category SMO Summary

Following from the above it becomes possible to put together a complete approach;

1. Select an initial two classes to consider denoted generically A and B.

2. From these two classes determine the two points with maximum and minimum values of $\mathbf{v}_B^T(A)\boldsymbol{\eta}$ as in equation (79). These will be denoted i_{up} and i_{low} respectively as they correspond to those associated with b_{up} and b_{low} in the work by Keerthi et al. (2001).

3. Perform updates as outlined in equations (74) and (76) until convergence by some criteria (*e.g.* updates are all below some threshold) is achieved. Point selection is made following the standard two loop approach of a `for` loop attempting to update all points and a `while` loop considering only non-extremal ones. Updates are attempted with respect to either i_{up} or i_{low} and these maximal and minimal points are updated after each iteration.

4. Once convergence has been achieved for these two classes then select another two classes and repeat steps 2 and 3. Do this until all possible combinations of classes have been attempted.

5. Repeat the entire process until no updates are made. At this point the Karush-Kuhn-Tucker (KKT) conditions should be very nearly satisfied, and should be checked to ensure that they are at least within some acceptable limit of satisfaction.

This is summarised in the pseudocode of Table 1.

This approach is clearly closely related to the structure of a pairwise coupling algorithm (Subsection 3.2) however now with a single optimisation problem as a focus. Clearly the algorithm may be more computationally intense than that of Kreßel (1999) for two reasons. First, each stage updates involves matrix multiplications instead of scalar ones. Second, as indicated by step 5, more than one pass may be required. On the other hand, there might be some reduction in overall iterations required in a particular class-class combination as each optimisation is not starting from 'scratch', rather updates from previous combinations may have had a positive impact.

Experimentally however it has been observed that the traditional pairwise coupling approach is more computationally efficient, and this has also been noted in the literature (Hsu & Lin, 2002; Rifkin & Klautau, 2004). As alluded to in Subsection 3.2, a combined approach is possible, which will be referred to as the combined pairwise, all-together algorithm. Broadly, this is as follows;

1. Perform the pairwise optimisation described by Kreßel (1999). This optimisation requires the implementation of $\frac{M(M-1)}{2}$ standard SV classifiers with the slight change that instead of using the standard 2-class ε value of 1, use the ε values corresponding to the particular pairwise optimisation.

2. Map the results into the classification spatial arrangement (Figures 1 or 2 for example). This can be done easily by observing that in the product $\mathbf{V}(\vartheta_i)\boldsymbol{\alpha}_i$ of equation (19) the element $\alpha_{i,\theta}$ multiplies $\mathbf{v}_\theta(\vartheta_i)$ — recall that this is perpendicular to the boundary between θ and ϑ_i. As such then the result of the binary classification optimisation between classes ϑ_i and θ can be used to directly provide the value $\alpha_{i,\theta}$. Note that the constraint in equation (17) is still satisfied.

3. Finalise the 'all-together' single optimisation following the steps outlined earlier in this Subsection.

In short the bulk of the optimisation is performed with the standard pairwise methodology. The geometrical approach detailed in Section 2 is used to manipulate the output such that a unified consistent result can be obtained with little additional computational effort. This has the clear advantage that a practitioner can be sure of exactly on what basis the classification is being made without having to resort to *ad hoc* heuristics.

8. Examples

Extensive investigations into comparative performance of multi-category SVM methods have been detailed by Hsu and Lin (2002), and they present current benchmark training times. As discussed, their work has found that pairwise coupling approaches are far more computationally efficient than others. This has also been found to be the case for the first SMO algorithm proposed in Subsection 7.4 and the main aim in this Section is to investigate the performance of the combined pairwise, 'all-together' algorithm. Both standard binary and the described multi-category SMO were coded in a straightforward way. No dynamic caching or low-level code refinements were used in this initial proof-of-concept investigation as it was felt that such detailed optimisations are best done together in a consistent way, as in the dedicated comparative work of Hsu and Lin (2002).

The datasets used were obtained from the University of California repository (Blake & Merz, 1998). For illustrative purposes the training and test output results on the DNA dataset are presented in Figure 9. Here it is clear to see how the pairwise result has been

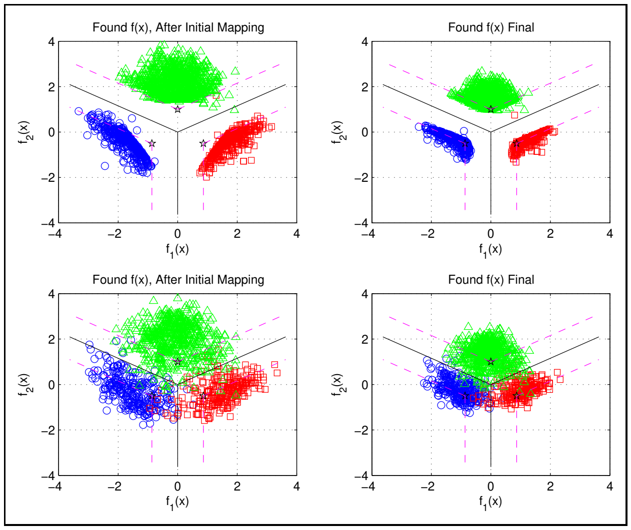


Figure 9: DNA data outputs for training and test data cases. *The mapped pairwise and optimised 'all-together' results are shown. Margins analogous to those in the two-class case are clearly visible and are shown by dashed lines. Training data forms the top row, test data the bottom. The more numerous, 'N' case is given by green triangles, 'EI' by blue circles, and 'IE' by red squares. The stars are indicative of the class target vectors.*

mapped into the classification plane of Figure 1, and what changes are made in performing the 'all-together' additional optimisation. In short the 'N' class appears to have intermingled a little more with the 'EI' class and less with the 'IE' class. As well the 'all-together' outputs fill the corners of the margin intersections more completely, while the pairwise outputs tend to cut them off. This has been often observed in other implementations.

The training time is heavily dependent on the tolerance to within which convergence is desired. This value, referred to as τ by Keerthi et al. (2001) indicates the variation allowed between b_{up} and b_{low} as discussed in Subsections 7.3 and 7.4. The effect of this has been additionally investigated for two values of τ, and the results are tabulated in Tables 2 and 3. In these experiments Gaussian kernels were used and appropriate values of σ and C were chosen by trial and error such that output accuracies (where accuracy refers to percentage classification error rate) of the 'all-together' implementation were comparable to those of Hsu and Lin (2002).

The actual accuracies recorded are given in the Table, however recall that, as noted in Section 3.1, the optimisation problem being solved is the generic 'all-together' one and, as such, judicious choices of σ and C should mean that *the same accuracy rates are achievable by all such algorithms*. Clearly as the implicit model behind the pairwise approach is slightly different it may indeed be able to achieve slightly different accuracy results. With this in mind the aim here has not been to incessantly tweak hyperparameters to achieve marginally superior results, but simply to look at the big picture of performance.

Problem	M	N	$\tau = 0.03C$			$\tau = 0.001C$		
			Pair	All	Alone	Pair	All	Alone
DNA	3	2000	0.8	1.1	1.5	1.1	3.7	11.7
Vehicle	4	766	0.4	2.7	5.3	0.5	3.5	3.9
Satimage	6	4435	3.0	10.8	41.8	3.6	9.0	27.6
Segment	7	2079	2.4	13.2	47.9	3.2	16.2	42.0
Vowel	11	891	0.7	3.5	13.3	1.0	18.5	22.8
Letter	26	15000	129.0	129.9	2119.2	142.3	1373.7	5573.4

Table 2: Optimisation times (seconds) for various example problems. *Columns present results obtained using the pairwise algorithm and the 'all-together' SMO algorithm discussed. In all cases 'Pair' refers to pairwise optimisation time results, meanwhile 'All' denotes additional refinement time* i.e. *that required to progress from the pairwise result to the 'all-together' result. Finally 'Alone' identifies time taken by the 'all-together' algorithm without initial pairwise optimisation.*

Problem	M	N	$\tau = 0.03C$		$\tau = 0.001C$	
			ER(Pair)	ER(All)	ER(Pair)	ER(All)
DNA	3	2000	4.4	4.6	4.6	4.5
Vehicle	4	766	15.0	18.8	17.5	20.0
Satimage	6	4435	10.6	10.8	9.7	9.2
Segment	7	2079	3.0	2.6	3.0	3.0
Vowel	11	891	3.0	3.0	3.0	3.0
Letter	26	15000	8.8	8.8	8.9	8.8

Table 3: Optimisation error rates (percentages) for various example problems. *Columns present experimentally obtained results using the pairwise and 'all-together' multi-category SMO algorithms discussed. 'ER(Pair)' refers to the test error rate of the pairwise method and 'ER(All)' to that of the 'all-together' algorithm.*

In continuing with this mindset, no class weightings were introduced, and target vectors were set to be equidistant. Clearly it may well be the case that these could actually be perturbed, and class weights introduced to improve performance, with no additional computational effort, however in this initial investigation this has not been done.

The experiments were all run on a 2.8GHz P4 with 1GB RAM[7]. From Tables 2 and 3 the following points become apparent,

1. The optimisation times presented here are of magnitudes similar to those of Hsu and Lin. Although it has not been the aim of this work to produce highly refined optimal code, and although such comparisons are always going to be problematic in terms of implementation specifics, this result is, in itself, positive. Generally, the most accurate implementation of the algorithm presented in the preceding sections (when $\tau = 0.001C$) convergence times are similar to those of Hsu and Lin for their 'all-together' implementation. Briefly, their optimisation times were; for DNA, 13.5s, for vehicle 88.6s, for satimage 48.2s, for segment 66.4s, for vowel 14.1s, and for letter 8786.2s. As such we consider the advantage obtained here through extra computational power as roughly equivalent to the effect of their extra coding.

7. Hsu and Lin (2002) had a 500MHz P3 with 384MB RAM.

It is worth noting that there is additional intrinsic value in the intuitiveness, flexibility and its ease of implementation of the presented algorithm, something the standard SMO algorithm is well known for. As highlighted, no additional computational effort is required to alter class regions or introduce class weights (Subsection 2.3), neither of which have been considered by Hsu and Lin (2002).

2. It is possible to approximately quantify the relative effect of combining the pairwise and 'all-together' algorithms in context. In short it typically halves them, although the variation on this is quite large. This result appears roughly consistent for both values of τ.

3. As anticipated, error rate results do not strongly favour the pairwise or 'all-together' methods; this is always going to be a case-by-case issue.

9. Conclusion

A geometric framework for understanding multi-category classification has been introduced, through which many existing 'all-together' algorithms can be understood. The structure allows the derivation of a parsimonious optimisation function, which is a direct extension of the binary SV classification optimisation function. This can be seen in that no special case considerations need be made in order that the mathematics reduce to the standard result when the number of classes, $M = 2$. Further, the framework enables considerable generalisation of the problem and incorporation of relative class knowledge without any additional computational complexity. As far as actual optimisation results are concerned, the virtues of the proposed framework, in fact, apply to the other 'all-together methods as well.

It has been found by Hsu and Lin (2002) and Rifkin and Klautau (2004), among others, that the pairwise SV method converges with a substantial speed advantage over existing multi-category methods. However pairwise results require some heuristic to combine them. This can be avoided by mapping them to the geometric framework described and 'fine-tuning' to obtain the consistent 'all-together' solution. This refining can be performed by any multi-category 'all-together' algorithm.

The ability of the framework to compare algorithms has been illustrated by a brief discussion of Fisher consistency. This has shown graphically illustrated how different loss structures compare and how most result in Fisher inconsistent optimisation problems.

Generalisation bounds have been derived with the aim of the framework presented which are tighter than those previously presented in the literature. These have also benefited from a simpler derivation than those previously presented due to the fact that well-known scalar methods developed for the two class case have been directly applicable. Previously there was a need to extend them to more cumbersome vector methods.

In addition to providing a more generic and flexible framework, this architecture may well provide insights regarding how to further improve on the speed of existing multi-category SV classification algorithms (whether coupled with a pairwise optimisation, or not). An initial example of how this might be achieved has been developed in the formulation of a straightforward multi-category SMO variant algorithm. The proof-of-concept experimental results have shown that this, combined with the mapping of pairwise results, is already comparable with the optimisation speeds achieved by Hsu and Lin (2002) in their benchmark work, despite the fact that their implementation code is highly refined and includes features such as dynamic caching. Future efforts based on the geometric framework described should be able to outperform existing standards.

References

Allwein, E. L., Schapire, R. E., & Singer, Y. (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, *1*(113-141).

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, *44*(2), 525–536.

Bartlett, P. L., Jordan, M. I., & McAuliffe, J. D. (2004). Large margin classifiers: Convex loss, low noise and convergence rates. *Advances in Neural Information Processing Systems*, *16*.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases.. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Blanz, V., Schölkopf, B., Bülthoff, H., Burges, C. J. C., Vapnik, V. N., & Vetter, T. (1996). Comparison of view-based object recognition algorithms using realistic 3D models. In von der Malsburg, C., von Seelen, W., Vorbrüggen, J. C., & Sendhoff, B. (Eds.), *Artificial Neural Networks*, Vol. 1112 of *Springer Lecture Notes in Computer Science*, pp. 251–256, Berlin.

Bredensteiner, E. J., & Bennett, K. P. (1999). Multicategory classification by support vector machines. *Computational Optimizations and Applications*, *12*, 53–79.

Crammer, K., & Singer, Y. (2001a). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*, 265–292.

Crammer, K., & Singer, Y. (2001b). Pranking with ranking. In *Advances in Neural Information Processing*, Vol. 14.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods* (1st edition). Cambridge University Press.

Dietterich, T., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

Elisseeff, A., Guermeur, Y., & Paugam-Moisy, H. (1999). Margin error and generalization capabilities of multi-class discriminant systems. Tech. rep. NC2-TR-1999-051-R, NeuroCOLT2.

Fung, G., & Mangasarian, O. L. (2001a). Multicategory proximal support vector machine classifiers. Tech. rep. 01-06, Data Mining Institute.

Fung, G., & Mangasarian, O. L. (2001b). Proximal support vector machine classifiers. In *Proceedings KDD-2001*, pp. 77–86, San Francisco.

Fürnkranz, J. (2002). Round robin classification. *Journal of Machine Learning*, *2*, 721–747.

Guermeur, Y. (2000). Combining discriminant models with new multi-class SVMs. Tech. rep., NeuroCOLT2.

Guermeur, Y. (2002). Combining discriminant models with new multi-class SVMs. *Pattern Analysis and Applications*, *5*, 168–179.

Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. In Michael, M. J. K., Jordan, I., & Solla, S. A. (Eds.), *Advances in Neural Information Processing*, Vol. 10, pp. 507–513. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Herbrich, R., Graepel, T., & Campbell, C. (2000a). Bayes point machines. *Journal of Machine Learning Research*, *1*, 245–279.

Herbrich, R., Graepel, T., & Obermayer, K. (2000b). Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, *1*, 115–132.

Hill, S. I. (2007). Notes on the generalisation performance and Fisher consistency of multicategory classifiers. Tech. rep. CUED/F-INFENG/TR.583, Engineering Dept, University of Cambridge.

Hill, S. I., & Doucet, A. (2005). A framework for kernel-based multi-category classification. Tech. rep. CUED/F-INFENG/TR.508, Engineering Dept., University of Cambridge.

Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, *13*, 415–425.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, *13*, 637–649.

Kindermann, J., Leopold, E., & Paaß, G. (2000). Multi-class classification with error correcting codes. In Leopold, E., & Kirsten, M. (Eds.), *Treffen der GI-Fachgruppe 1.1.3 Maschinelles Lernen*. GMD Report 114.

Kreßel, U. H.-G. (1999). Pairwise classification and support vector machines. In Schölkopf, B., Burges, C. J. C., & Smola, A. J. (Eds.), *Advances in Kernel Methods: Support Vector Learning*. MIT Press.

Lee, Y., Lin, Y., & Wahba, G. (2001). Multicategory support vector machines. Tech. rep. 1043, Department of Statistics, University of Wisconsin.

Lee, Y., Lin, Y., & Wahba, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, *99*, 659–672.

Mangasarian, O. L., & Musicant, D. R. (2001). Lagrangian support vector machines. *Journal of Machine Learning Research*, *1*, 161–177.

Mayoraz, E., & Alpaydın, E. (1999). Support vector machines for multi-class classification. In *Proceedings of the International Workshop on Artifical Neural Networks (IWANN99)*.

Paugam-Moisy, H., Elisseeff, A., & Guermeur, Y. (2000). Generalization performance of multiclass discriminant models..

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C. J. C., & Smola, A. J. (Eds.), *Advances in Kernel Methods - Support Vector Learning*, pp. 185–208. MIT Press, Cambridge, MA.

Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (2000). Large margin DAGs for multiclass classification. In Solla, S. A., Lean, T. K., & Müller, K.-R. (Eds.), *Advances in Neural Information Processing*, Vol. 12, pp. 547–553. MIT Press.

Rennie, J. D. M., & Rifkin, R. (2001). Improving multiclass text classification with the support vector machine. Memo AIM-2001-026, Massachusetts Institute of Technology Artificial Intelligence Laboratory.

Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, *5*, 101–141.

Schölkopf, B., Burges, C. J. C., & Vapnik, V. N. (1995). Extracting support data for a given task. In Fayyad, U. M., & Uthurusamy, R. (Eds.), *Proceedings, First International Conference on Knowledge Discovery and Data Mining*, pp. 252–257, Menlo Park, CA. AAAI Press.

Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Sebald, D. J. (2000). *Nonlinear Signal Processing for Digital Communications using Support Vector Machines and a New Form of Adaptive Decision Feedback Equalizer*. Ph.D. thesis, University of Wisconsin-Madison.

Sebald, D. J., & Bucklew, J. A. (2001). Support vector machines and the multiple hypothesis test problem. *IEEE Transactions on Signal Processing*, *49*(11), 2865–2872.

Sejnowski, T. J., & Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text.. *Journal of Complex Systems*, *1*, 145–168.

Suykens, J. A. K., & Vandewalle, J. (1999). Multiclass least squares support vector machines. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'99)*, Washington DC, USA.

Tewari, A., & Bartlett, P. L. (2007). On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, *8*, 1007–1025.

Van Gestel, T., Suykens, J. A. K., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B., & Vandewalle, J. (2001). Benchmarking least squares support vector machine classifiers. *Machine Learning*, *54*(1), 5–32.

Van Gestel, T., Suykens, J. A. K., Lanckriet, G., Lambrechts, A., De Moor, B., & Vandewalle, J. (2002). Multiclass LS-SVMs: Moderated outputs and coding-decoding schemes. *Neural Processing Letters*, *15*, 45–58.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.

Wang, F., Vuurpijl, L. G., & Schomaker, L. R. B. (2000). Support vector machines for the classification of western handwritten capitals. In Schomaker, L. R. B., & Vuurpijl, L. G. (Eds.), *Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition*, pp. 167–176.

Weston, J. A. E. (1999). *Extensions to the Support Vector Method*. Ph.D. thesis, University of London.

Weston, J. A. E., & Watkins, C. (1999). Support vector machines for multi-class pattern recognition. In *Proceedings of the 7th European Symposium On Artificial Neural Networks*.

Williamson, R. C., Smola, A. J., & Schölkopf, B. (2001). Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, *47*(6), 2516–2532.

Zhang, T. (2004a). An infinity-sample theory for multi-category large margin classification. *Advances in Neural Information Processing*, *16*.

Zhang, T. (2004b). Statistical analysis of some multi-category large margin classification. *Journal of Machine Learning Research*, *5*, 1225–1251.