The Benefits of Modelling Slack Variables in SVMs

Fengzhen Tang[1], Peter Tiňo[1], Pedro Antonio Gutiérrez[2] and Huanhuan Chen[3]

[1] School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK (email:fxt126, P.Tino@cs.bham.ac.uk).

[2] Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba 14071, Spain (email:pagutierrez@uco.es).

[3] UBRI, School of Computer Science and Technology, University of Science and Technology of China, Hefei, 230027, China (email: hchen@ustc.edu.cn).

Keywords: Learning using privileged information, ordinal regression, Slack variable modelling, support vector ordinal regression.

Abstract

In this paper, we explore the idea of modelling slack variables in Support Vector Machine (SVM) approaches. The study is motivated by SVM+, which models the slacks

through a smooth correcting function that is determined by additional (privileged) information about the training examples not available in the test phase. In this paper we take a closer look at the meaning and consequences of smooth modelling of slacks, as opposed to determining them in an unconstrained manner through the SVM optimization programme. To better understand this difference we only allow the determination and modelling of slack values on the same information - i.e. using the same training input in the original input space. We also explore whether it is possible to improve classification performance by combining (in a convex combination) the original SVM slacks with the modelled ones. We show experimentally that this approach not only leads to improved generalization performance, but also yields more compact, lower complexity models. Finally we extend this idea to the context of ordinal regression, where a natural order among the classes exists. The experimental results confirm principal findings from the binary case.

1 Introduction

Support Vector Machines (SVMs) have gained wide popularity over the last decades. They were shown to be effective for many problems on numerous applications such as digit recognition, face detection, speaker identification and so on (Burges, 1998). For binary classification, SVMs construct a separating hyperplane as the decision boundary to separate the positive examples from the negative ones with maximum margin. To deal with the case of overlapping classes, SVM formulation utilizes non-negative 'slack' variables to tolerate misclassification in training data. Non-linear class separa-

tion structure can be addressed through so-called ‘kernel trick’ - kernels map the input data into a higher dimensional feature space where linear separation hyperplane can be applied.

Recently, Vapnik and Vashist (2009) extended SVM framework to SVM+ by modelling the slack variables of training points through so-called correcting functions to incorporate an additional privileged information. The privileged information is available for inputs during training stage but unavailable in the test phase. Modelling slacks using privileged information is feasible, as the slacks are only used in the training stage, but not in the test phase. SVM+ can achieve superior performance when compared to standard SVM trained without privileged information (Vapnik and Vashist, 2009).

In this paper, we explore the benefits of modelling slack variables in SVM from a different perspective. We study the difference between determining the slack values as in the original SVM and modelling them via a smooth correcting function. For a systematic study of this issue we need to make sure that the determination and modelling of slack values are done using the same information - i.e. using the same training examples in the original input space. In other words, to obtain model-based slack values we will employ the SVM+ model, but the domain of the correcting functions will be the original input space, rather than the privileged information one.

Having obtained two sets of slack values on the same problem (i.e. the ones obtained through a standard SVM optimization procedure and the ones obtained from the correcting function), we further investigate in a data driven manner which kind of slack construction is more preferable for the given problem. To that end we consider a new set of slack values obtained as a convex combination of the ‘standard’ and model based

slacks. The values of mixing coefficients in the convex combination indicate the preferred slack construction. We will refer to this approach as *SVMvP* and introduce a principled (but costly), as well as a practical algorithm to implement this idea.

Ordinal regression problems are multi-class classification problems where a natural order among categories can be observed (Cardoso and Pinto da Costa, 2007) and they are recently receiving considerable attention (Lin and Li, 2012; Fouad and Tiňo, 2012; Sánchez-Monedero et al., 2013; Seah et al., 2012). We extend the idea of modelling slacks to ordinal regression problems on the basis of the Support Vector Ordinal Regression with IMplicit constraints (SVORIM) (Chu and Keerthi, 2005). SVORIM constructs multiple parallel hyperplanes separating the adjacent classes (in the class order), by stipulating that each hyperplane separates all the points in higher classes from all the points in lower classes. As we do for binary SVM, we first model the slack variables for each hyperplane using a correcting function (*SVORIMP*). We then derive a method based on combining the slacks from the standard optimization procedure of SVORIM and the slacks from the correcting functions with a mixing parameter v (*SVORIMvP*).

In summary, this paper explores the idea of modelling slack variables in SVM framework. The contributions of this paper are three-fold:

- We investigate the differences between modelling slacks and obtaining their values in an unconstrained manner through the optimization programme. This is done by slack modelling via a correcting function using the original information, instead of privileged information.

- We introduce methodologies for obtaining slacks through a convex combination of model based and optimized slack values, which, as will be shown, leads to

lower model complexity and enhanced generalization performance.

- We extend these ideas to the case of ordered classes in the framework of ordinal regression.

The paper has the following structure: Section 2 briefly describes SVM and SVM+. The idea of modelling slack variables using original training inputs is presented in Section 3 and *SVMvP* is demonstrated in Section 3.1. Section 4 extends the idea of modelling slack variables to ordinal regression. Section 5 presents the experimental results and analysis. The main findings are discussed and summarized in Section 6.

2 Background

2.1 Support Vector Machine (SVM)

In this section we briefly review SVMs for classification problems (for more details see e.g. Vapnik, 1998; Burges, 1998).

Given a training set of l examples, represented by input-output pairs $(\boldsymbol{x}_i, y_i)$, $\boldsymbol{x}_i \in R^n$, $y_i \in \{-1, 1\}$, the aim is to construct a decision boundary (separating hyperplane) that separates positive examples from the negative ones with maximum margin. This can be formulated as the following constrained optimization problem:

$$\begin{aligned} &\min \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 + C \sum_{i=1}^{l} \xi_i \\ &\text{s. t. } \; y_i(\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, \; \xi_i \geq 0, \; \forall \, i. \end{aligned} \tag{1}$$

where $||\boldsymbol{w}||$ is the Euclidean norm of $\boldsymbol{w}$ and $\Phi(\cdot)$ is the feature mapping induced by the associated kernel $\mathcal{K}(\cdot, \cdot)$. Non-negative slack variables ξ_i are utilized to relax the

constraints and allow some misclassification. $C \geq 0$ is a hyper-parameter chosen by user. This problem is usually transformed to its dual according to the Karush-Kuhn-Tucker (KKT) conditions (Boyd and Vandenberghe, 2004):

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\text{s. t. } \sum_{i=1}^{l} \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C,\ \forall\, i. \end{aligned} \tag{2}$$

Once optimal α's are obtained, the decision function for a new input vector $\boldsymbol{x}$ is given by:

$$F(\boldsymbol{x}) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b\right). \tag{3}$$

2.2 Learning Using Privileged Information in SVM framework

Learning Using Privileged Information (LUPI) (Vapnik and Vashist, 2009), also known as Learning Using Hidden Information (Vapnik et al., 2009), has been introduced to deal with situations where additional (privileged) information $\boldsymbol{x}^* \in X^*$ about training examples $\boldsymbol{x} \in X$ is known during training but is unavailable in the test phase. Privileged information appears in several application domains (Vapnik and Vashist, 2009; Vapnik et al., 2009), for example, in the time series prediction, privileged information is the behaviour of the time series in the future; in cancer prediction using biopsy images, the privileged information is the pathologist's report etc.

An extension of SVM learning algorithm, known as SVM+, has been suggested as a candidate for LUPI in (Vapnik and Vashist, 2009; Vapnik et al., 2009). In SVM+, the slack variables for inputs in X are determined by a correcting function operating in the

privileged space X^*,

$$\xi(\boldsymbol{x}^*) = \boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}^*) + b^*,$$

where Φ^* is the feature map induced by the kernel operating on X^*. Replacing the slacks in (1) by the slack variable model defined above, the problem becomes:

$$\min_{\boldsymbol{w},\boldsymbol{w}^*,b,d} \frac{1}{2}\left[\| \boldsymbol{w} \|^2 + \gamma(\| \boldsymbol{w}^* \|^2)\right] + C\sum_{i=1}^{l}[\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i^*) + d]$$

$$\text{s. t.} \tag{4}$$

$$y_i[\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b] \geq 1 - [\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i^*) + d],\ \forall\, i,$$

$$\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i^*) + d \geq 0,\ \forall\, i.$$

where γ is a hyper-parameter used to control the capacity for the correcting function in X^* space. The Lagrangian reads:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{w}^*, b, d, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}[\| \boldsymbol{w} \|^2 + \gamma(\| \boldsymbol{w}^* \|^2)] + C\sum_{i=1}^{l}[\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i^*) + d] \tag{5}$$

$$-\sum_{i=1}^{l}\alpha_i\{y_i[\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b] - 1 + [\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i^*) + d]\} - \sum_{i=1}^{l}\beta_i[\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i^*) + d],$$

with the corresponding dual:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$-\frac{1}{2\gamma}\sum_{i,j=1}^{l}(\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C)\mathcal{K}^*(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*)$$

$$\text{s. t.} \tag{6}$$

$$\sum_{i=1}^{l}(\alpha_i + \beta_i - C) = 0,\ \sum_{i=1}^{l} y_i\alpha_i = 0,\ \alpha_i \geq 0,\ \beta_i \geq 0,\ \forall\, i.$$

$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\mathcal{K}^*(\boldsymbol{x}_i^*, \boldsymbol{x}_j^*)$ are kernels in X and X^* spaces, respectively. SVM+ have been successfully used on a variety of data sets with privileged information, (e.g. Ribeiro et al., 2010; Liang and Cherkassky, 2007).

2.3 SVM for Ordinal Regression - SVORIM

Support Vector Ordinal Regression with IMplicit constraints (SVORIM) (Chu and Keerthi, 2005) is a generalization of the binary SVM (Vapnik and Lerner, 1963; Vapnik, 1998; Burges, 1998; Chang and Lin, 2010; Kotsiantis et al., 2006) to learning to rank or ordinal regression. While the key concept of the SVM classifier is to construct a hyperplane separating the positive examples from the negative ones with maximum margin, SVORIM classifier extends this idea by constructing multiple parallel hyperplanes separating the adjacent classes (in the class order). In contrast to Support Vector Ordinal Regression with EXplicit constraints (SVOREX) (also proposed by Chu and Keerthi (2005) by enforcing an order on the adjacent thresholds explicitly), SVORIM ensures the threshold order implicitly by stipulating that the j-th hyperplane (corresponding to threshold b_j) separates all points from classes $\leq j$ from all points of classes $> j$.

Consider an ordered set of classes $\{1, 2, ..., J\}$. In SVORIM (Chu and Keerthi, 2005), there are two sets of slack variables $\boldsymbol{\xi}$ and $\boldsymbol{\nu}$ and the primal problem is formulated as follows:

$$\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\nu}} \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 + C \sum_{j=1}^{J-1} \left(\sum_{k=1}^{j} \sum_{i=1}^{n^k} \xi_{ki}^j + \sum_{k=j+1}^{J} \sum_{i=1}^{n^k} \nu_{ki}^j \right),$$

s.t. (7)

$$\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i^k) - b_j \leq -1 + \xi_{ki}^j,\ \xi_{ki}^j \geq 0, k = 1, ..., j \text{ and } i = 1, ..., n^k,$$

$$\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i^k) - b_j \geq +1 - \nu_{ki}^j,\ \nu_{ki}^j \geq 0, k = j+1, ..., J,\ i = 1, ..., n^k.$$

where j runs over 1 to $J-1$. ξ_{ki}^j and ν_{ki}^j are the 'left' and 'right' slacks, respectively, for the i-th point in class k with respect to the separating hyperplane between classes j and $j+1$ and n^k is the number of patterns of class k.

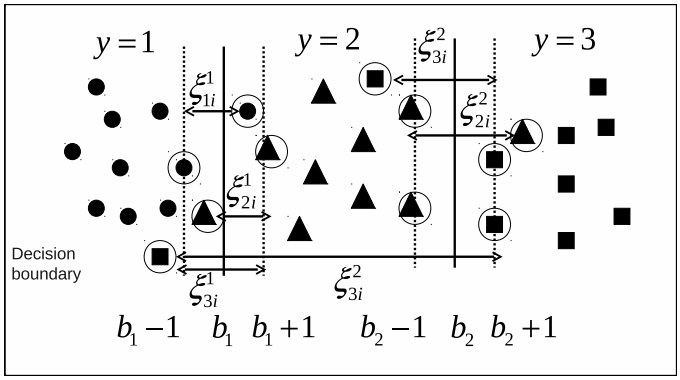


Figure 1: Illustration of SVORIM. All the examples are mapped to their function values $\boldsymbol{w} \cdot \Phi(\boldsymbol{x})$ along the horizontal axis.

Note that since in SVORIM there is a slack variable for each *(data point, decision boundary)* pair, there is no need to have different notations for the 'left' and 'right' slacks, $\boldsymbol{\xi}$ and $\boldsymbol{\nu}$, respectively. The left-right slacks are necessary for the explicit constraint formulation, but not for the implicit one. The idea of SVORIM can be summarized in Figure 1: for a threshold b_j, the function values $\boldsymbol{w} \cdot \Phi(\boldsymbol{x})$ of all examples from all the lower categories should be less than the lower margin $b_j - 1$ and the function values $\boldsymbol{w} \cdot \Phi(\boldsymbol{x})$ of all examples from all the upper categories should be greater than the upper margin $b_j + 1$. Slacks of each example with respect to every threshold are allowed to relax the constraints.

3 Modelling Slack Variables in SVM classification

Vapnik and Vashist (2009) have theoretically and empirically justified the idea of modelling the slack variables using privileged information (SVM+). If the idea of modelling slack variables (as opposed to obtaining their individual values through optimization problem) is reasonable, then it makes sense to ask what happens if we build a slack

variable model using the original information. In other words, we would like to analyse the modelling approach for determining slacks by imposing $X = X^*$ and using the SVM+ framework:

$$\xi_i = \xi(\boldsymbol{x}_i) = \boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x_i}) + d. \tag{8}$$

The proposed model, which is denoted by *SVMP*, formulates the slack model as kernel regression and can thus be naturally incorporated into the SVM framework. The decision rule and the correcting function are found by solving the following constrained optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{w}^*,b,d} \frac{1}{2}[\| \boldsymbol{w} \|^2 + \gamma(\| \boldsymbol{w}^* \|^2)] + C\sum_{i=1}^{l}[\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i) + d] \\ \text{s. t.} \quad & \\ &y_i[\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b] \geq 1 - [\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i) + d],\ \forall\, i, \\ &\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i) + d \geq 0,\ \forall\, i. \end{aligned} \tag{9}$$

The Lagrangian is constructed as in eq (5). By applying KKT conditions, we can obtain an optimization problem only depending on α's and β's:

$$\begin{aligned} &\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i\alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &-\frac{1}{2\gamma}\sum_{i,j=1}^{l}(\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C)\mathcal{K}^*(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s. t.} \quad & \\ &\sum_{i=1}^{l}(\alpha_i + \beta_i - C) = 0,\ \sum_{i=1}^{l} y_i\alpha_i = 0,\ \alpha_i \geq 0, \beta_i \geq 0,\ \forall\, i. \end{aligned} \tag{10}$$

Once the optimal α's and β's are obtained, the decision function has the same form as in eq. (3) and the corresponding correcting function reads:

$$\xi(\boldsymbol{x}) = \frac{1}{\gamma}\sum_{i=1}^{l}(\alpha_i + \beta_i - C)K^*(\boldsymbol{x}_i, \boldsymbol{x}) + d, \tag{11}$$

where the bias d is computed from $\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i) + d = 0$, for any training point with $\beta_i > 0$ and we take the average over all such points. The bias b in the decision function can be computed from any point whose corresponding multiplier α_i is greater than zero from $y_i[\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b] - 1 + \xi(\boldsymbol{x}_i) = 0$ (we take the average over all such points).

In the standard SVM construction the slacks are not constrained by any smooth model, but are determined directly in the optimization procedure. We have shown how the slack values can be obtained in a model based manner through correcting functions. Next, we will combine the two kinds of slacks in a convex combination.

3.1 Convex Combination of Model Based and Optimized Slack Values (*SVMvP*)

In this section we propose to use slack values obtained from a convex combination of the slacks obtained in the SVM and *SVMP* frameworks. This proposal allows slack values to be moved between modelled slacks and independently learned slacks so that we can answer in a data-driven way what kind of slack values is preferable for a given task. This idea can be formulated as follows:

$$r_i = (1 - v)\xi_i + v\xi(\boldsymbol{x}_i), \quad v \in [0, 1], \quad \forall\, i. \tag{12}$$

We refer to the model operating with slacks r_i as *SMPvP*.

Given v, using the slacks r_i in (12), the problem can be formulated as:

$$\min_{\boldsymbol{w},\boldsymbol{w}^*,b,d,\boldsymbol{\xi}} \frac{1}{2}[\| \boldsymbol{w} \|^2 + \gamma(\| \boldsymbol{w}^* \|^2)] + C\sum_{i=1}^{l} r_i$$

s. t.

$$y_i[\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b] \geq 1 - r_i, \ \forall\, i,$$

$$\xi_i \geq 0, \ \boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i) + d \geq 0, \ \forall i.$$

As before, we construct the Lagrangian:

$$\begin{aligned}\mathcal{L} = {} & \frac{1}{2}[\| \boldsymbol{w} \|^2 + \gamma(\| \boldsymbol{w}^* \|^2)] + C\sum_{i=1}^{l}\{(1-v)\xi_i + v[\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i) + d] \\ & + \sum_{i=1}^{l}\alpha_i\{1 - (1-v)\xi_i - v[\boldsymbol{w}^* \cdot \Phi^*(\boldsymbol{x}_i) + d] + y_i[\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b]\} \\ & - \sum_{i=1}^{l}\beta_i(\Phi^*(\boldsymbol{x}_i) + d) - \sum_{i=1}^{l}\theta_i\xi_i\end{aligned} \tag{13}$$

where α_i, β_i and θ_i are non-negative Lagrangian multipliers. Again, we can transform the problem into its dual:

$$\begin{aligned}\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} & \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ & -\frac{1}{2\gamma}\sum_{i,j=1}^{l}(v\alpha_i + \beta_i - vC)(v\alpha_j + \beta_j - vC)\mathcal{K}^*(\boldsymbol{x}_i, \boldsymbol{x}_j)\end{aligned}$$

s. t.

$$\sum_{i=1}^{l}(v\alpha_i + \beta_i - vC) = 0, \ \sum_{i=1}^{l} y_i\alpha_i = 0, \ 0 \leq \alpha_i \leq C, \ \beta_i \geq 0, \ \forall\, i.$$

Once the optimal α's and β's are obtained, we can use the following KKT complementary conditions

$$\beta_i(\boldsymbol{w}^*\Phi^*(x_i) + d) = 0, \tag{14}$$

$$\alpha_i\{y_i[\boldsymbol{w} \cdot \Phi(x_i) + b] - 1 + r_i\} = 0, \tag{15}$$

$$\theta_i\xi_i = 0, \tag{16}$$

to compute the bias $d = -\boldsymbol{w}^* \Phi^*(\boldsymbol{x}_i)$ using any $\boldsymbol{x}_i$ for which $\beta_i \neq 0$. We take the average over all such points. Once we have the bias d of the slack model, we can compute bias b of the decision function through Equation (15), using any $\boldsymbol{x}_i$ for which $0 < \alpha_i < C$. Again, we take the average over all such points.

The model introduced above has 5 hyper-parameters that need to be tuned (e.g. via cross-validation), namely, kernel widths of the decision and slack model (correcting) functions, σ and σ^*, respectively, regularization parameters C and γ, and coefficient v of the slack convex combination. In practice, we obtain the slacks ξ_i in standard SVM and model slacks $\xi(\boldsymbol{x}_i)$ by running *SVMP*, respectively. Having slacks ξ_i, $\xi(\boldsymbol{x}_i)$ and combination coefficient v, we compute the new slacks r_i and recover the corresponding decision boundary from the SVM model formulation by solving (r_i are fixed):

$$\begin{aligned} &\min \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 \\ &\text{s. t. } y_i\{\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b\} \geq +1 - r_i, \ \forall\, i. \end{aligned} \tag{17}$$

The Lagrangian has the following form:

$$\mathcal{L} = \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 - \sum_{i=1}^{l} \alpha_i \{y_i[\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i) + b] - 1 + r_i\}, \tag{18}$$

The solution requires the following conditions to be met:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{l} \alpha_i y_i \Phi(\boldsymbol{x}_i) = 0, \tag{19}$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0. \tag{20}$$

By substituting the solution of (19) - (20) into (18), we obtain the corresponding dual

problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{l}(1-r_i)\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x}_j)$$

s. t.

$$\sum_{i=1}^{l}\alpha_i y_i = 0, \tag{21}$$

$$\alpha_i \geq 0,\ \forall\, i.$$

After finding α's, the decision function is obtained as in standard SVM (section 2).

4 Modelling Slacks in SVM based Ordinal Regression

In this section, we extend the idea of modelling slacks in binary SVM to Support Vector Ordinal Regression with IMplicit constraints (SVORIM) (Chu and Keerthi, 2005).

We chose SVORIM instead of the explicit one in Chu and Keerthi (2005), since in SVOREX the j-th hyperplane ($j = 1, 2, ...J-1$, where J is the number of classes) is constrained only by the slacks of patterns from adjacent classes, whereas in SVORIM it is constrained by the slacks of patterns from all classes. As the key aspect of our method is modelling of slacks, the SVORIM framework can provide more flexibility through greater number of correcting functions.

In this section, we present the detailed derivation of the *SVORIMP* algorithm, which models slack variables for each threshold b_j by a correcting function, as follows:

$$\xi^j(\boldsymbol{x}) = \boldsymbol{w}_j^* \Phi^*(\boldsymbol{x}) + d_j, \tag{22}$$

where $j = 1, 2, ..., J-1$. Replacing the slack variables by the slack models (22) and considering the primal in (7), we can formulate the following primal problem using

correcting functions:

$$\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{w}^*,\boldsymbol{d}} \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 + \frac{\gamma}{2} \sum_{j=1}^{J-1} (\parallel \boldsymbol{w}_j^* \parallel^2) + C \sum_{j=1}^{J-1} \sum_{k=1}^{J} \sum_{i=1}^{n^k} (\boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j),$$

$$s.t. \text{ for every } j = 1, ..., J-1, \tag{23}$$

$$\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i^k) - b_j \leq -1 + (\boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j), \text{for } k = 1, ..., j \text{ and } i = 1, ..., n^k,$$

$$\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i^k) - b_j \geq +1 - (\boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j), \text{for } k = j+1, ..., J \text{ and } i = 1, ..., n^k,$$

$$\boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j \geq 0. \; \forall \; i, \; j, \; k.$$

As in the previous sections, we construct the Lagrangian:

$$\begin{aligned}
\mathcal{L} &= \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2}\|\boldsymbol{w}_j^*\|^2 + C \sum_{j=1}^{J-1} \sum_{k=1}^{J} \sum_{i=1}^{n^k} (\boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j) \\
&- \sum_{j=1}^{J-1} \sum_{k=1}^{j} \sum_{i=1}^{n^k} \{\alpha_{ki}^j (-1 + \boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j - \boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i^k) + b_j)\} \\
&- \sum_{j=1}^{J-1} \sum_{k=j+1}^{J} \sum_{i=1}^{n^k} \alpha_{ki}^j (-1 + \boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j + \boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i^k) - b_j) \\
&- \sum_{j=1}^{J-1} \sum_{k=1}^{J} \sum_{i=1}^{n^k} \beta_{ki}^j (\boldsymbol{w}_j^* \cdot \Phi^*(\boldsymbol{x}_i^k) + d_j),
\end{aligned} \tag{24}$$

where α_{ki}^j and β_{ki}^j are non-negative multipliers. The KKT conditions for the primal problem require the following conditions hold true:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{w} + \sum_{j=1}^{J-1} \sum_{k=1}^{j} \sum_{i=1}^{n^k} \alpha_{ki}^j \Phi(\boldsymbol{x}_i^k) - \sum_{j=1}^{J-1} \sum_{k=j+1}^{J} \sum_{i=1}^{n^k} \alpha_{ki}^j \Phi(\boldsymbol{x}_i^k) = 0, \tag{25}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_j^*} &= \gamma \boldsymbol{w}_j^* + C \sum_{k=1}^{J} \sum_{i=1}^{n^k} \Phi^*(\boldsymbol{x}_i^k) - \sum_{k=1}^{j} \sum_{i=1}^{n^k} \alpha_{ki}^j \Phi^*(\boldsymbol{x}_i^k) \\
&- \sum_{k=j+1}^{J} \sum_{i=1}^{n^k} \alpha_{ki}^j \Phi^*(\boldsymbol{x}_i^k) - \sum_{k=1}^{J} \sum_{i=1}^{n^k} \beta_{ki}^j \Phi^*(\boldsymbol{x}_i^k) = 0,
\end{aligned} \tag{26}$$

$$\frac{\partial \mathcal{L}}{\partial b_j} = -\sum_{k=1}^{j} \sum_{i=1}^{n^k} \alpha_{ki}^j + \sum_{k=j+1}^{J} \sum_{i=1}^{n^k} \alpha_{ki}^j = 0, \; \forall \; j, \tag{27}$$

$$\frac{\partial \mathcal{L}}{\partial d_j} = \sum_{k=1}^{J} \sum_{i=1}^{n^k} (\alpha_{ki}^j + \beta_{ki}^j - C) = 0, \forall j. \tag{28}$$

By substituting the solutions of (25)-(28) into (24), we have the following dual problem:

$$\begin{aligned}
&\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{k,i}(\sum_{j=1}^{J-1}\alpha_{ki}^{j}) \\
-\quad &\frac{1}{2}\sum_{k,i}\sum_{k',i'}\left\{(\sum_{j=1}^{k-1}\alpha_{ki}^{j}-\sum_{j=k}^{J-1}\alpha_{ki}^{j})(\sum_{j=1}^{k'-1}\alpha_{k'i'}^{j}-\sum_{j=k'}^{J-1}\alpha_{k'i'}^{j})\mathcal{K}(\boldsymbol{x}_i^k,\boldsymbol{x}_{i'}^{k'})\right\} \\
-\quad &\frac{1}{2\gamma}\sum_{j=1}^{J-1}\sum_{k,i}\sum_{k',i}\left\{(\alpha_{ki}^{j}+\beta_{ki}^{j}-C)(\alpha_{k'i'}^{j}+\beta_{k'i'}^{j}-C)\mathcal{K}(\boldsymbol{x}_i^k,\boldsymbol{x}_{i'}^{k'})\right\} \qquad (29)
\end{aligned}$$

s. t.

$$\begin{aligned}
&\sum_{k=1}^{j}\sum_{i=1}^{n^k}\alpha_{ki}^{j}=\sum_{k=j+1}^{J}\sum_{i=1}^{n^k}\alpha_{ki}^{j},\forall j, \\
&\sum_{k=1}^{J}\sum_{i=1}^{n^k}(\alpha_{ki}^{j}+\beta_{ki}^{j}-C)=0,\forall j, \\
&\alpha_{ki}^{j}\geq 0,\beta_{ki}^{j}\geq 0,\forall i,\forall j.
\end{aligned}$$

Once the solution of the dual problem is found, the value of discriminant function at a new input $\boldsymbol{x}$ is:

$$F(\boldsymbol{x})=\sum_{k,i}(\sum_{j=1}^{k-1}\alpha_{ki}^{j}-\sum_{j=k}^{J-1}\alpha_{ki}^{j})\mathcal{K}(\boldsymbol{x}_i^k,\boldsymbol{x}). \qquad (30)$$

The correcting functions for each threshold have the form,

$$\xi^j(\boldsymbol{x})=f_j(\boldsymbol{x})+d_j, \qquad (31)$$

where $f_j(\boldsymbol{x})=\frac{1}{\gamma}\sum_{k=1}^{J}\sum_{i=1}^{n^k}(\alpha_{ki}^{j}+\beta_{ki}^{j}-C)\mathcal{K}^*(\boldsymbol{x}_i^k,\boldsymbol{x})$, and the bias d_j is computed by averaging over $-f_j(\boldsymbol{x}_i^k)$ for all the points which $\beta_{ki}^{j}>0$, $j=1,...,J-1$. The threshold b_j can be computed by any $\alpha_{ki}^{j}>0$, in the following way:

$$b_j=\begin{cases}F(\boldsymbol{x}_i^k)+1-\xi^j(\boldsymbol{x}_i^k) & k\leq j,\\ F(\boldsymbol{x}_i^k)-1+\xi^j(\boldsymbol{x}_i^k) & k>j.\end{cases} \qquad (32)$$

The threshold is taken the average for these points. Then, the predictive ordinal decision function is defined as:

$$\arg\min_i F(\boldsymbol{x})<b_i. \qquad (33)$$

The time complexity of this algorithm is $O((J-1)^3 l^3)$, and there are four hyper-parameters need to be tuned.

4.1 Convex Combination of Model Based and Optimized Slack values in SVORIM (*SVORIMvP*)

This section demonstrates the algorithm denoted by *SVORIMvP*, which uses slack values from convex combination of slack values obtained from the correcting functions and values from the standard SVORIM optimization procedure, as follows:

$$r_{ki}^{j} = (1-v)\xi_{ki}^{j} + v\xi_{j}(\boldsymbol{x}_{i}^{k}), \tag{34}$$

where the mixing weight $0 \leq v \leq 1$ can be tuned through cross-validation. We obtain the slacks ξ_{ki}^{j} and $\xi_{j}(\boldsymbol{x}_{i}^{k})$ by running SVORIM and *SVORIMP*, respectively. After determining the combined slacks (34), the primal problem is formulated as:

$$\begin{aligned}
&\min_{\boldsymbol{w},\boldsymbol{b}} \frac{1}{2} \parallel \boldsymbol{w} \parallel^2 \\
&s.t. \\
&\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_{i}^{k}) - b_j \leq -1 + r_{ki}^{j}, k = 1, ..., j \text{ and } i = 1, ..., n^k, \\
&\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_{i}^{k}) - b_j \geq +1 - r_{ki}^{j}, k = j+1, ..., J,\ i = 1, ..., n^k,
\end{aligned} \tag{35}$$

where $j = 1, ..., J-1$ and the corresponding dual can be formulated as:

$$\begin{aligned}
&\max_{\boldsymbol{\alpha}} \sum_{k,i} (\sum_{j=1}^{J-1} \alpha_{ki}^{j}(1 - r_{ki}^{j})) \\
- \quad &\frac{1}{2} \sum_{k,i} \sum_{k',i'} \{(\sum_{j=1}^{k-1} \alpha_{ki}^{j} - \sum_{j=k}^{J-1} \alpha_{ki}^{j})(\sum_{j=1}^{k'-1} \alpha_{k'i'}^{j} - \sum_{j=k'}^{J-1} \alpha_{k'i'}^{j}) \mathcal{K}(\boldsymbol{x}_i^k, \boldsymbol{x}_{i'}^{k'})\} \\
&s.t. \\
&\sum_{k=1}^{j} \sum_{i=1}^{n^k} \alpha_{ki}^{j} = \sum_{k=j+1}^{J} \sum_{i=1}^{n^k} \alpha_{ki}^{j}, \ \forall\, j, \\
&\alpha_{ki}^{j} \geq 0, \ \forall\, i, \ j, \ k.
\end{aligned}$$

Once the solution for dual has been obtained, the threshold can be computed by any $\alpha_{ki}^{j} > 0$ as:

$$b_j = \begin{cases} F(\boldsymbol{x}_i^k) + 1 - r_{ki}^{j} & k \leq j, \\ F(\boldsymbol{x}_i^k) - 1 + r_{ki}^{j} & k > j. \end{cases} \tag{36}$$

The time complexity of this algorithm remains $O((J-1)^3 l^3)$. Compared to *SVORIMP*, *SVORIMvP* has one more hyper-parameters. However, by using the same trick as we do for *SVMvP*, model fitting of *SVORIMvP* only costs the effort of the same order as that of *SVORIMP*.

5 Experimental Results and Analysis

We evaluated our methodology on several data sets of different nature and origin. The input vectors were normalized to zero mean and unit variance. RBF kernels were used both in classifier design and in slack variable modelling with kernel widths σ and σ^*, respectively, except the case of synthetic data for linear decision boundary where a linear kernel was used in the classifier design.

In our experiment, the ranges allowed for the parameters were as following: $\sigma \in \{0.1, 0.5, 1, 5, 10\}$, $\sigma^* \in \{0.1, 0.5, 1, 5, 10\}$, $C \in \{1, 10, 50, 100, 500\}$, $\gamma \in \{0.001, 0.01, 0.1, 1, 10, 100, 500, 1000\}$ and the value of mixing coefficient v for unconstrained and model based slacks was taken from $\{0, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 1\}$. Hyper-parameters were tuned via grid search based on 5-fold cross validation over the training set. We used the **cvx** Matlab tool[1] as optimization routine to training the SVM based algorithms mentioned in this paper using SDPT3 solver. Denoting the number of training examples by l, for SVM, *SVMP* and *SVMvP* the time complexity is $O(l^3)$, $O((2l)^3)$ and $Q((l + 2l + l)^3)$, respectively.

We first present and discuss experiments on binary classification. We employed 2 synthetic, 10 benchmark datasets and a very large dataset. We then move on to ordinal regression, where 4 real world time series data sets were used.

5.1 Binary Classification

Synthetic Data Toy experiments were performed to evaluate the proposed algorithm using randomly generated two-dimensional data from class-conditional Gaussian distributions with diagonal covariance matrix. In each experiment there were 2 classes and we randomly and independently generated 100 training and 2000 testing points per class. The data generation and model fitting/evaluation process was repeated 10 times. Tables 1 and 2 contain the mean (and StDev) results over the 10 trails.

In the first experiment, both classes shared the same spherical covariance structure (identity matrix I), meaning that the optimal separation boundary was linear. The means

[1] http://cvxr.com/cvx

of positive and negative classes were set to $(1, 1)$ and $(-1, -1)$, respectively. The ideal separation line goes through the origin with directional vector $(1, -1)$. We employed linear kernels $\mathcal{K}$ and Gaussian kernels $\mathcal{K}^*$ in *SVMP* and *SVMvP*, respectively.

In the second experiment, the three algorithms were tested on data with non-linear optimal decision boundary. The class-conditional means remained the same, while the covariance structure of the negative and positive classes was $2I$ and I, respectively. The decision boundary 'bends' towards the positive class.

Table 1: Classification error for synthetic datasets

Decision boundary	SVMvP	SVM	SVMP	SVMvP ($v = 1$)
Linear	0.0762±0.0030	0.0782 ± 0.0032	0.0762 ± 0.0027	0.0769±0.0029
Non-linear	0.1367 ±0.0056	0.1398±0.0053	0.1353±0.0052	0.1372±0.0062

Table 2: Number of support vectors for synthetic datasets

Decision boundary	SVMvP	SVM	SVMP	SVMvP ($v = 1$)
Linear	10.60±16.04	39.20 ± 10.65	151.40 ± 77.42	17.3±24.47
Non-linear	60.30±46.49	78.80±13.24	158.60±44.45	96.80±36.22

Table 1 summarizes classification performance of the three models in the two synthetic data experiments. In addition we also report results for the *SVMvP* model with v set to 1[2]. Note that the *SVMvP* model with $v = 1$ is not equivalent to the *SVMP* model, although both use model based slacks only. This is because in the *SVMvP* model the decision boundary is reconstructed from the slacks as described in section 3.1. However,

[2] We are thankful to the anonymous reviewer for making this suggestion.

it can be shown that when $v = 0$, the *SVMvP* model is identical to the original SVM. The number of support vectors in each model is recorded in table 2.

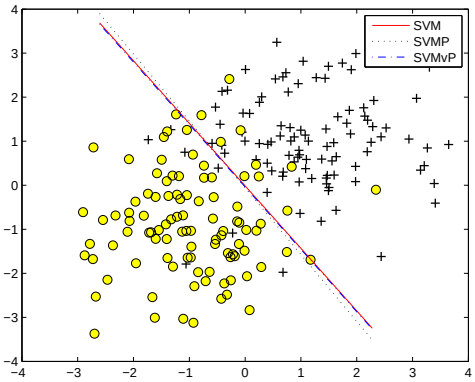
Test errors of *SVMvP* and *SVMP* were slightly smaller than that of SVM. Compared to SVM, the number of *SVMP* support vectors was much larger, while the number of support vectors of *SVMvP* was much smaller than in the case of SVM. *SVMvP* with $v = 1$ achieve similar (but slightly inferior) performance to *SVMvP* with v as a free parameter. However, the model complexity of *SVMvP* is lower than that of *SVMvP* with $v = 1$.

As an example, we show in Figure 2 separation lines (a) and support vectors of SVM (b), *SVMP* (c) and *SVMvP* (d), for one trial in the first experiment. It appears that *SVMvP* needs much less support vectors to determine the separating line. Analogous results were found for data with non-linear separation in the second experiment (see Figure 3).

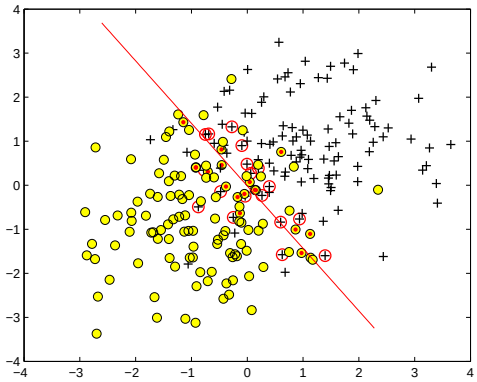
The values of mixing parameter v for slacks selected through cross-validation in the first and second experiment were (mean $\pm$ StDev) 0.84 ± 0.2665 and 0.83 ± 0.2406, respectively. In the two experiments, the methodology prefers model-based slacks[3].

Benchmark datasets 10 benchmark datasets from the UCI repository (Asuncion and Newman, 2007) were used to evaluate the three methods. The datasets are briefly described in Table 3. Each data set was randomly and independently partitioned into training/test splits 100 times, yielding 100 re-sampled training/test sets. In

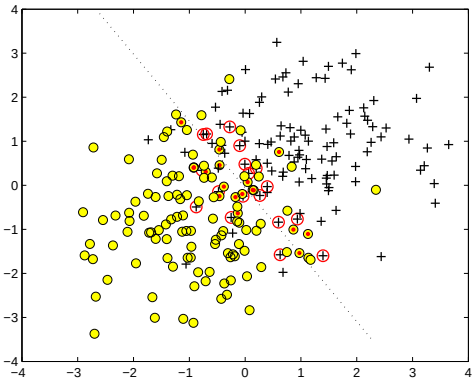
[3] As mentioned earlier, by imposing $v = 0$, *SVMvP* becomes standard SVM.



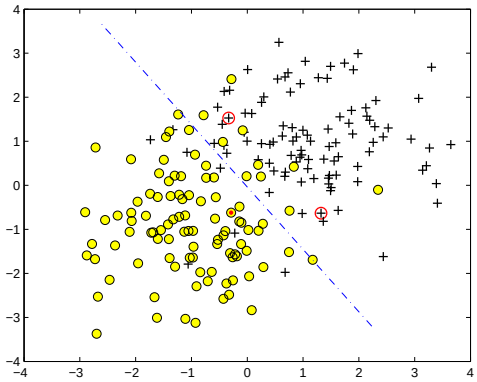
(a) Decision boundary of the three algorithms



(b) SVM, # support vectors is 36.



(c) SVMP, # support vectors is 37



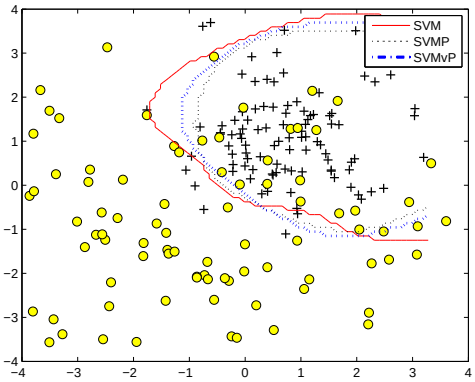
(d) SVMvP, # support vectors is 3

Figure 2: Illustration of linear decision boundary. Black $+$ represents positive examples while yellow $\circ$ describes negative examples. Support vectors from positive examples are red circled, while support vectors from negative examples are marked with red dot in the centre.

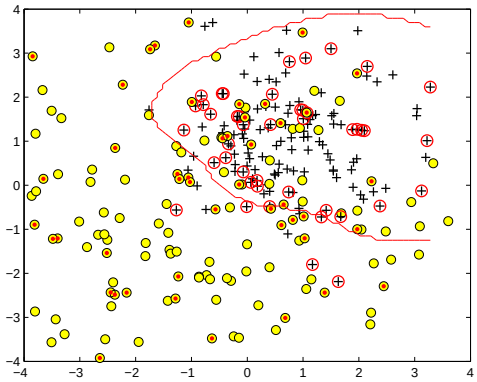
addition, we also employed a large dataset (*Covertype*[4]) containing 536301 data items[5]. We randomly partitioned the *Covertype* set into 600 (disjoint) folds. The models were fitted and tested on the first 6 folds - in particular, the first fold was used for training,

[4] We are thankful to the anonymous reviewer for making this suggestion.

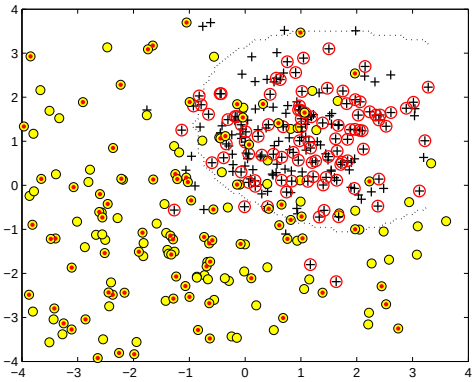
[5] after removing items with missing values



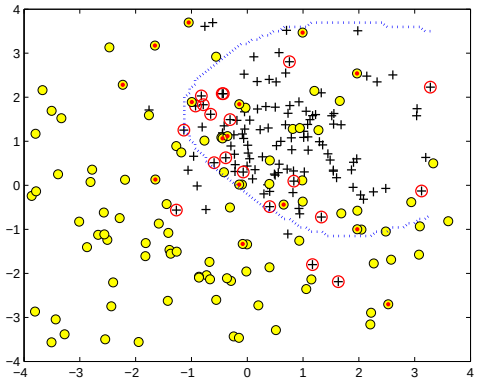
(a) Decision boundary of the three algorithms



(b) SVM, # support vectors is 102.



(c) SVMP,# support vectors is 151.



(d) SVMvP, # support vectors is 36.

Figure 3: Illustration of non-linear decision boundary. Support vectors are marked in the same as in Figure 2.

the remaining 5 folds for testing. The procedure was then repeated on the next block of 6 folds, and so on, until all 100 6-fold blocks were used.

Tables 4 and 5 report the average performance on the data sets over the 100 trails. The classification error of *SVMP* was consistently smaller than that of SVM. However, the number of support vectors was mainly (10 cases out of 11) greater than for SVM. *SVMvP* achieved slightly worse classification error compared to *SVMP*, but still better than SVM. The support vector set of *SVMvP* was significantly smaller than that of both

SVM and *SVMP*. As in the synthetic data experiments, *SVMvP* with $v = 1$ achieved comparable, but slightly worse performance than *SVMvP* with free v, while compared with *SVMvP*, the number of support vectors in *SVMvP* with $v = 1$ was higher.

Table 3: Description of the benchmark datasets. m is dimensionality of the input vector.

Dataset	Cancer	Diabetes	Heart	Solar	Thyroid	German	Australian	Breast cancer	Fourclass	Liver disorders
m	9	8	13	9	5	20	14	10	2	6
# training / # test	132/131	384/384	135/135	72/72	107/108	500/500	345/345	342/341	431/431	173/172

Table 6 summarizes statistical differences between the methods using Wilcoxon test (Wilcoxon, 1945). The significance level was set to $\alpha = 0.1$. For this analysis we considered the benchmark, as well as the synthetic data sets (total of 13 data sets). Each entry of the table reports the number of datasets for which the row method beat the column method in the statistically significant manner (wins), the number of datasets where the differences were not statistically significant (draws) and the number of datasets where the row method performed significantly worse than the column method (loses). We also included *SVMvP* with $v = 1$ for comparison purposes. *SVMvP* and *SVMP* obtained statistically better classification error than SVM for 12 datasets, while *SVMvP(v=1)* for 10 datasets. With respect to the number of support vectors, *SVMvP* had statistically significantly smaller support vector sets than *SVMP* and SVM for 13 and 11 datasets, respectively. Fixing $v = 1$ statistically increased the error of SVMvP for 8 datasets and the number of support vectors for 8. Moreover, *SVMvP(v=1)* was only able to improve number of support vectors with respect to SVM for 7 datasets and it was beaten by SVM in 2 datasets. The tests confirm that the results previously observed in Tables 4 and 5, refuting that the differences could have been obtained by chance.

The values of parameter v selected from cross-validation in *SVMvP* are given in Table 7. It is interesting to observe that for all studied data sets the mixing of slack values is biased towards the model-based slacks provided by the correcting function.

These experiments indicate that modelling the slack variables using (8) has a potential to improve generalization performance, at the cost of increased model complexity. However, using convex combination of unconstrained and model-based slacks (12) can result in superior model of significantly reduced complexity.

Discussion and Analysis Our experimental results show that *SVMP* can improve generalization performance over SVM at the expense of increased model complexity. The i-th training point is considered as support vector if its corresponding α_i value is positive. Therefore, in SVM the points on the hyperplanes $\boldsymbol{w} \cdot \Phi(\boldsymbol{x}) + b = -1$ and $\boldsymbol{w} \cdot \Phi(\boldsymbol{x}) + b = 1$, together with the points whose corresponding slack value are bigger than zero are support vectors. Hence, the determination of slack values will influence the number of support vectors. Slacks in SVM are obtained independently through optimization programme, whereas the slacks in *SVMP* change according to a smooth correcting function. Points in the neighbourhood of an input with a positive slack will tend to have positive slacks imposed by the model. This can result in an increased number of support vectors, when compared with SVM.

From our experimental results we see that the classification boundary reconstruction from slacks used in *SVMvP* decreases the number of support vectors. Comparing the dual problems for SVM and *SVMvP*, (2)-(3) and (21)-(21), respectively, we notice two principal differences. First, the term $\sum_i^l \alpha_i$ in SVM is replaced by $\sum_i^l \delta_i \alpha_i$, $\delta_i = 1 - r_i$,

in *SVMvP*. Second, α_i in *SVMvP* are no longer bounded by the penalty C (as in SVM).

If $r_i > 1$, meaning that the corresponding input $\boldsymbol{x}_i$ is on the wrong side of the boundary, the weight δ_i of α_i in (21) is negative, forcing α_i to zero (or 'small' value). At the same time, the term

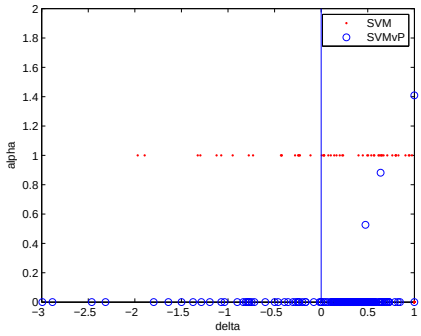
$$-\sum_{i}^{l}\sum_{j}^{l}\alpha_i\alpha_j y_i y_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

is encouraging high α values for points considered similar under the kernel $\mathcal{K}$ (e.g. spatially close under a Gaussian kernel), but with different class labels. The overall effect in *SVMvP* is that a smaller number of points on the correct side of the decision boundary, but close to it, will have high α values, whereas the other points will have small, or vanishing α's. This is illustrated in Figure 4. *SVMvP* model is usually much more sparse than the standard SVM. Unlike in SVM (dots), the support vectors with non-zero α's in *SVMvP* (circles) are predominantly located on the correct side of the decision boundary ($\delta_i = 1 - r_i > 0$) and attain much higher values.

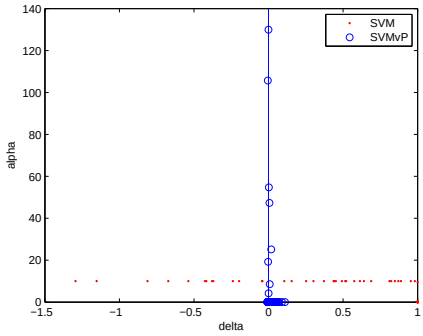
5.2 Ordinal Regression

In this section, we present the experimental results on modelling slacks in SVORIM. We employed time series data sets (see Table 8), which were quantized into a series of categories with natural order, so they can be tackled as ordinal regression problems.

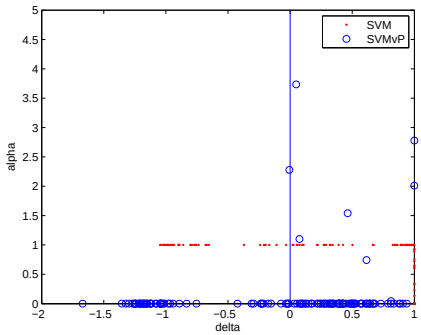
Four different time series have been considered. *Sunspot* is the annual sunspot numbers from 1700-1988. *Fish* data contains 453 monthly values of estimated fish recruitment in the period 1950-1987. *Wine* data set contains Australian red wine sales in the period of 1980-1991. Finally, *Birth* data set contains births per 10,000 of 23 year old women in U.S. in the period of 1917-1975. For each of the four time series $\{s_t\}$, a new



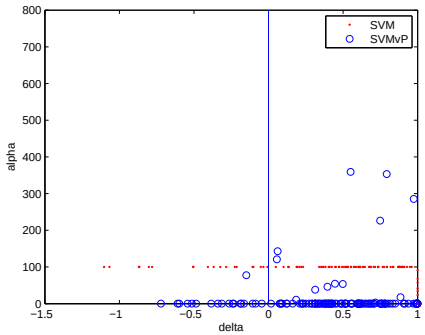
(a) Synthetic dataset for linear decision boundary.



(b) Synthetic dataset for non-linear decision boundary .



(c) Cancer dataset.



(d) Heart dataset.

Figure 4: Distribution of the multipliers α_i and the weights δ_i for two synthetic and two real datasets.

series of differences $D_t = s_t - s_{t-1}$ was created and was then quantized into a symbolic stream $\{y_t\}$ through:

$$y_t = \begin{cases} 1 \text{ (extreme down)} & \text{if } D_t < \theta_1 < 0 \\ 2 \text{ (normal down)} & \text{if } \theta_1 \leq D_t < 0 \\ 3 \text{ (normal up)} & \text{if } 0 \leq D_t < \theta_2 \\ 4 \text{ (extreme up)} & \text{if } \theta_2 \leq D_t \end{cases}$$

The cut values θ_1, θ_2 were chosen so that classes 1, 2, 3 and 4 contain 10%, 40%, 40% and 10% of sequence elements D_t. We used the values of the previous 5 time steps as

input features. We randomly split these datasets into training and test set 5 times. The final results are the average results over the 5 trails.

The zero/one classification errors are given in Table 9, the mean absolute errors[6] are given in Table 10 and the number of support vectors is listed in Table 11. The results of the Wilcoxon tests are given in Table 12. According to Tables 9, 10 and 11, the classification error of *SVORIMP* is much smaller than that of SVORIM but the number of support vectors of *SVORIMP* is slightly greater than that of SVORIM. The classification error of *SVORIMvP* is more or less the same as *SVORIMP* but the number of support vectors is much smaller than that of both SVORIM and *SVORIMP*. Thus, as in the binary case, modelling slack variables in SVORIM using original information can improve the generalization performance of the learner and decrease the model complexity. Finally, Table 13 includes the values of v selected by cross-validation. Again, a trend similar to the binary case can be observed – *SVORIMvP* tends to select the values from the correcting functions, although the original slacks can also play an important role. Finally, as in the previous experiments, in general, *SVORIMvP* with $v = 1$ tend to yield comparable, or slightly worse performance than *SVORIMvP* with free v. When compared with *SVORIMvP*, the number of support vectors in *SVORIMvP* with $v = 1$ tends to be higher.

[6] The average difference between the predicted and target classes in terms of the number of categories separating them in the ordinal scale.

6 Discussion and Conclusion

In the framework of Learning with Privileged Information, Vapnik and Vashist (2009) proposed to incorporate privileged information through modelling the SVM slack variables through a smooth correcting function whose domain is the privileged space. This is reasonable, since the correcting function/slacks are updated only in the training (model fitting) phase and are never used in the test phase. Indeed, as shown in (Pechyony and Vapnik, 2010), such an incorporation of additional information can lead to faster convergence (as the training sample size grows) to the true (optimal Bayes) model, provided the privileged information is 'informative enough' about the structure of the classification problem[7].

In this contribution we took a closer look at the meaning and consequences of (smooth) modelling of slacks, as opposed to determining them in an unconstrained manner through the SVM optimization programme. To investigate this issue, we asked: What is the difference between determining the slack values as in the original SVM and modelling them via a smooth function? To gain a better understanding of this difference we allowed the determination and modelling of slack values to be done using the same information – i.e. using the same training sample in the original input space. We then moved further and asked: Is it possible to improve classification performance by combining (in a convex combination) the original SVM slacks with the modelled ones? By checking the mixing weights we could determine in a data driven manner which of the two approaches to slack value determination are preferable for a given data set.

[7] Here, informative enough means that the correcting functions operating in the privileged space can provide slack values 'close' to the 'ideal' oracle slack values corresponding to the true underlying model.

We first introduced *SVMP*, which models the slack variables through a smooth correcting function in the original space. We introduced a principled method for convex mixing of the original and modelled slack values. However, the method needed tuning of five hyper-parameters. Therefore, we considered a more practical method, which obtains the original values ξ_i by running SVM and the model values $\xi(\boldsymbol{x}_i)$ by running *SVMP*. Those values are then combined and the decision boundary is recovered from the mixed slack values. Experimental results show that, compared with SVM, this approach (*SVMvP*) can lead to reduction in both the misclassification rate and the model complexity. Interestingly enough, for most data sets the modelled slacks were preferred (had higher mixing weight) to the original ones.

We then extended the idea of model based slacks to ordinal regression in the framework of SVORIM. We chose SVORIM instead of the explicit one (Chu and Keerthi, 2005), because the SVORIM framework can provide more flexibility for correcting function modelling through greater number of slacks. As for SVM, we first model slacks corresponding to each separating hyperplane using a correcting function (*SVORIMP*). Then we propose to use convex combination of the values ξ^j_{ki} obtained from SVORIM and the values $\xi^j(x^k_i)$ obtained from *SVORIMP*. The experimental results show that modelling slacks, as opposed to their determination as in the original SVORIM, improves the generalization performance and reduces the model complexity.

References

Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

Cardoso, J. S. and Pinto da Costa, J. F. (2007). Learning to classify ordinal data: The data replication method. *Journal of Machine Learning Research*, 8:1393–1429.

Chang, C. C. and Lin, C. J. (2010). LIBSVM: A library for support vector machines. *ACM Transactions Intelligent System Technology*, 2:1–27.

Chu, W. and Keerthi, S. S. (2005). New approaches to support vector ordinal regression. In *Proceedings of the 22nd International Conference on Machine learning*, (ICML'05), pages 145–152.

Fouad, S. and Tiňo, P. (2012). Adaptive metric learning vector quantization for ordinal classification. *Neural Computation*, 24(11):2825–2851.

Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E. (2006). Machine learning : a review of classification and combining techniques. *Artificial Intelligence*, 26:159–190.

Liang, L. and Cherkassky, V. (2007). Learning using structured data : Application to fMRI data analysis. In *Proceeding of International Joint Conference on Neural Networks*, pages 495–499.

Lin, H.-T. and Li, L. (2012). Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367.

Pechyony, D. and Vapnik, V. (2010). On the theory of learning with privileged information. In *Advances in Neural Information Processing Systems 23*.

Ribeiro, B., Silva, C., Vieira, A., Gaspar-Cunha, A., and das Neves, J. C. (2010). Financial distress model prediction using svm+. In *Proceeding of International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

Sánchez-Monedero, J., Gutiérrez, P. A., Tiňo, P., and Hervás-Martínez, C. (2013). Exploitation of pairwise class distances for ordinal classification. *Neural Computation*, 25(9):2450–2485.

Seah, C.-W., Tsang, I. W., and Ong, Y.-S. (2012). Transductive ordinal regression. *IEEE Transactionss on Neural Networks and Learning Systems*, 23(7):1074–1086.

Vapnik, V. (1998). *Statistical Learning Theory*. New York ; Chichester : Wiley.

Vapnik, V. and Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24(6):774–780.

Vapnik, V. and Vashist, A. (2009). A new paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557.

Vapnik, V., Vashist, A., and Pavlovitch, N. (2009). Learning using hidden information (learning with teacher). In *Proceeding of International Joint Conference on Neural Networks*, pages 3188–3195.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Table 4: Classification error for benchmark datasets

dataset	SVMvP	SVM	SVMP	SVMvP ($v = 1$)
Cancer	0.2360±0.0229	0.2504±0.0232	0.2356±0.0233	0.2406±0.0221
Diabetes	0.2157±0.016	0.2237±0.0163	0.2155±0.0152	0.2171±0.0160
Heart	0.1381±0.0182	0.1470±0.0189	0.1370±0.0189	0.1393±0.0183
Solar	0.3150±0.0341	0.3418±0.0386	0.3050±0.0317	0.3143±0.0346
Thyroid	0.0197±0.0151	0.0319±0.015	0.0177±0.0121	0.0207±0.0157
German	0.2286 ±0.0143	0.2372±0.0133	0.2263±0.0134	0.2295±0.0136
Australian	0.1194±0.0114	0.1300±0.0130	0.1186±0.0111	0.1202±0.0113
Breast cancer	0.0240±0.0065	0.0273±0.0065	0.0223±0.0058	0.0244±0.0067
Fourclass	0.000±0.0000	0.0001±0.0003	0.0000±0.0000	0±0.0000
Liver disorders	0.2562±0.0220	0.2733±0.0260	0.2540±0.0222	0.2582±0.0222
Covertype	0.2532 ± 0.0075	0.2549 ± 0.0075	0.2535± 0.0074	0.2537±0.0074

Table 5: Number of support vectors for benchmark datasets

dataset	SVMvP	SVM	SVMP	SVMvP(v = 1)
Cancer	67.51±27.94	79.33±10.12	113.04±21.08	83.69±26.61
Diabetes	173.53±102.12	218.85±23.4	323.33±59.70	210.92±94.29
Heart	46.2±33.4	80.3±19.49	109.03±29.15	65.63±31.76
Solar	30.19±23.27	45.16±5.39	68.45± 10.78	31.68±21.20
Thyroid	18.38±13.33	29.95±16.22	52.79±42.82	18.33±12.37
German	221.08±129.18	289.79±18.26	459.43±66.74	256.05±131.46
Australian	160.13±70.67	165.28±48.99	266.28±70.55	198.36±64.84
Breast cancer	32.84±28.22	51.19±21.26	212.32±139.04	41.29±31.25
Fourclass	18.53±4.49	28±27.82	19.16±4.67	19.79±11.29
Liver disorders	80.22±42.11	119.66±14.06	156.51±20.72	93.25±48.20
Covertype	409.49±79.07	539.55±28.14	841.76±91.37	413.23±97.74

Table 6: Results of Wilcoxon test for a significance level $\alpha = 0.1$ (Wins/Draws/Loses, i.e. number of datasets where the method of the row is significantly better than the method of the column, no significant differences can be found and it is significantly worse, respectively).

	Classification error (Wins/Draws/Loses)			# of SVs (Wins/Draws/Loses)		
Method	SVM	SVMP	SVMvP(v=1)	SVM	SVMP	SVMvP(v=1)
SVMvP	12/1/0	0/7/6	8/5/0	11/2/0	13/0/0	8/5/0
SVM	-	0/1/12	0/1/12	-	12/0/1	2/4/7
SVMP	-	-	10/3/0	-	-	1/0/12

Table 7: Optimal value of the slacks mixing parameter v

dataset	Cancer	Diabetes	Heart	Solar	Thyroid	German	Australian	Breast cancer	Fourclass	Liver disorders	Covertype
v	0.7861 ±0.2413	0.8545 ±0.2076	0.7477 ±0.3007	0.6936 ± 0.4199	0.8775 ±0.2420	0.8138 ±0.2711	0.8600 ±0.2033	0.8157 ±0.2730	0.9336 ±0.1626	0.8376 ±0.2642	0.7173 ± 0.3189

Table 8: Description of the ordinal datasets. m is the dimensionality of the input vector.

Dataset	Sunspot	Fish	Wine	Birth
m	5	5	5	5
# class	4	4	4	4
# training/ # test	222/56	265/177	118/13	40/8

Table 9: Classification error on ordinal datasets

Dataset	SVORIMvP	SVORIM	SVORIMP	SVORIMvP ($v = 1$)
Sunspot	0.3418 ±0.0459	0.4277±0.0549	0.3381±0.0452	0.4061 ±0.0602
Fish	0.5130 ± 0.0109	0.5571 ± 0.0267	0.5107±0.0117	0.5186±0.0152
Wine	0.3599±0.1032	0.4456±0.0788	0.3599±0.0896	0.4975±0.1079
Birth	0.3000±0.0685	0.4250±0.1118	0.3250±0.0685	0.3250±0.0685

Table 10: Mean Absolute Error on ordinal datasets

Dataset	SVORIMvP	SVORIM	SVORIMP	SVORIMvP ($v = 1$)
Sunspot	0.3851± 0.0699	0.4710±0.0765	0.3813±0.0652	0.4564± 0.0794
Fish	0.5819±0.0384	0.6271±0.0543	0.5684±0.0267	0.5774±0.0251
Wine	0.4291±0.1219	0.5819±0.0995	0.4214±0.1116	0.6150±0.1658
Birth	0.4000±0.0559	0.5500±0.1425	0.4500±0.1118	0.4250±0.0685

Table 11: Support vector size on ordinal datasets

Dataset	SVORIMvP	SVORIM	SVORIMP	SVORIMvP ($v = 1$)
Sunspot	147.8 ±52.77	191.40±10.23	206.20±29.78	221.20±2.16
Fish	69.40±50.34	235.80±9.86	265.00±0.00	181.80±106.06
Wine	64.7±46.81	108.2±5.35	115.4±5.21	117.20±1.68
Birth	23.80±14.60	37.20±2.59	37.60±1.82	20.80±10.99

Table 12: Results of Wilcoxon test on ordinal datasets for a significance level $\alpha = 0.1$ (Wins/Draws/Loses, i.e. number of datasets where the method of the row is significantly better than the method of the column, no significant differences can be found and it is significantly worse, respectively).

	Classification error (Wins/Draws/Loses)		
Method	SVORIM	SVORIMP	SVORIMvP(v=1)
SVORIMvP	0/4/0	0/4/0	1/3/0
SVORIM	-	0/4/0	0/4/0
SVORIMP	-	-	1/3/0
	Mean Absolute Error (Wins/Draws/Loses)		
Method	SVORIM	SVORIMP	SVORIMvP(v=1)
SVORIMvP	1/3/0	0/4/0	1/3/0
SVORIM	-	0/3/1	0/4/0
SVORIMP	-	-	1/3/0
	# of SVs (Wins/Draws/Loses)		
Method	SVORIM	SVORIMP	SVORIMvP(v=1)
SVORIMvP	0/4/0	1/3/0	0/4/0
SVORIM	-	1/3/0	0/4/0
SVORIMP	-	-	0/3/1

Table 13: Optimal value of the slack mixing parameter v in SVORIMvP

Dataset	Sunspot	Fish	Wine	Birth
v	0.8900±0.1342	0.7900±0.2460	0.9300 ±0.1304	0.7900±0.2748