# From circular ordinal regression to multilabel classification

Dieter Devlaminck[1], Willem Waegeman[2], Bruno Bauwens[1], Bart Wyns[1],
Patrick Santens[3], and Georges Otte[4]

[1] Department of Electrical Energy, Systems and Automation, Ghent University,
Technologiepark 913, 9052 Zwijnaarde, Gent, Belgium
{dieter.devlaminck@ugent.be}
[2] Department of Applied Mathematics, Biometrics and Process Control,
Ghent University, Coupure links 653, B-9000 Ghent, Belgium
[3] Department of Neurology, Ghent University Hospital,
De Pintelaan 185, 9000 Ghent, Belgium
[4] P.C. Dr. Guislain,
Fr. Ferrerlaan 88A, 9000 Ghent, Belgium

**Abstract.** Several applications domains like wind forecasting in meteorology and robot control in robotics demand for learning algorithms that are able to make discrete directional predictions. We refer to this problem setting as circular ordinal regression, since it shares the same properties as traditional ordinal regression, namely the need for a specific model structure and order-preserving loss functions. This article gives a detailed introduction to the topic and proposes two methods. The first one is a circular support vector approach (cSVM), parameterized with only two vectors. The second method converts circular ordinal regression to a multilabel classification approach that takes the circular ranking into account by minimizing the Hamming loss. We also present initial empirical results based on two toy examples and a real-life application in the area of brain-computer interfaces.

**Keywords:** circular ordinal regression, multilabel classification, brain-computer interface (BCI)

## 1 Introduction

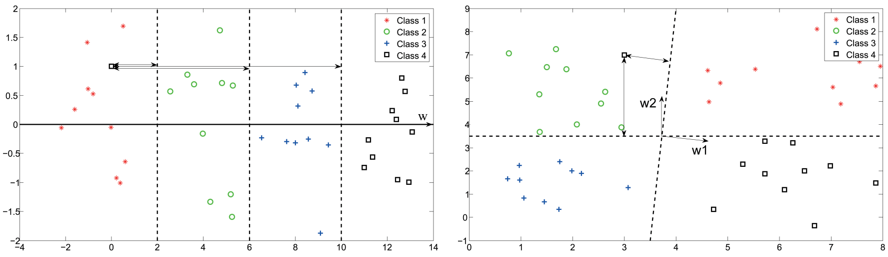
Wind forecasting and robot control are two domains where circular ordinal regression methods are needed, but potential applications arise in all learning settings where (discrete) directional predictions have to be made. Think in this context at automatic control of vehicles or aircrafts. In the field of brain-computer interfaces as well, where the goal consists of detecting patterns from brain signals, circular ordinal regression problems are appearing, for steering wheelchairs of disabled people, or for cursor control in computer games. From our point of view, circular ordinal regression methods should be very related to traditional ordinal regression methods, which are nowadays commonly used in social sciences, medicine and information retrieval, for example, to learn people's preferences, but also in many other disciplines.

Choosing an appropriate machine learning algorithm for a given application depends on many factors. Two factors that play a major role in this choice are the given structure of the data and the loss function of interest, and mainly these two factors characterize the need for developing specific algorithms in the field of ordinal regression [16]. Although ordinal regression contains elements of both classification and regression, there are some noteworthy differences. Consider, for example, the task of giving grades A>B>C>D to students. When judging upon the students work, it is far worse to rate a student with a C, when he actually deserves an A, than rating him with a B. This type of ordering information is not taken into account in ordinary multi-class classification problems. So, clearly ordinal regression is not the same as classification, but is it really regression? The answer is no, because there is no real metric defined between the ordinal scales of the target variables. Moreover, the target variables are discrete in contrast to classical regression. A straightforward naive approach for ordinal regression consists of converting the ordinal target variables to a numeric scale and applying a classical regression method. Other simple ideas rely on aggregating several binary classifiers [9, 12], by nesting these classifiers in a way that preserves the order of the classes. Many more advanced ordinal regression algorithms have been presented in recent years; neural network approaches have been considered in [2, 4, 11], but also support vector machines have been modified for the purpose of ordinal regression [15, 3].

Circular ordinal regression, as described above, assumes that the ranking is circular. Returning to the example of grading students, we could consider the following circular order instead, A>B>C>D>A (off course, this ranking does not make sense in the problem of grading people). Nevertheless, this kind of ordering will be the topic of the paper and occurs in applications where one has to estimate directions. Here, we apply it to data of the Brain-Computer Interface (BCI) competition with the goal of driving a wheel chair based on brain signals. Section 2 starts with a brief and general discussion of the ordinal model and presents the circular SVM (cSVM) approach for any number of classes based on the idea presented in [8]. Section 3 describes how the circular order can be obtained by encoding the original labels as several binary classification problems, so that the original problem becomes a binary relevance multilabel classification problem, because the circular mean absolute error coincides with the multilabel Hamming loss. The section also shows an additional connection with the cSVM model in case of four classes. Section 4 discusses the results on some data sets.

## 2 Circular ordinal regression using two hyperplanes

Most algorithms for binary classification consider as model structure a linear or nonlinear discriminant or ranking function, which is fitted to the data. Usually, this discriminant function can be represented as a linear real-valued function in



(a) (b)

**Fig. 1.** A visualization of the traditional ordinal regression model (a) and the proposed circular regression approach (b). The two main differences are the structure of the data and the number of parameter vectors to be fit. In the circular regression case we need two vectors, defining the two decision boundaries, instead of only one in the ordinal regression case.

a given feature space

$$f : \mathcal{X} \mapsto \Re : \mathbf{x} \mapsto f(\mathbf{x}) = \mathbf{w}^T \cdot \Phi(\mathbf{x})\,, \tag{1}$$

with $\mathbf{w}$ a vector of parameters, $\mathcal{X}$ the input domain and $\Phi$ a transformation of that input vector to feature space. From this discriminant function a binary classifier is typically derived by taking the sign. An easy way to extend these binary classifiers to multiple classes is by constructing an ensemble of such classifiers.

For some applications, like medical decision making, regular multiclass models are inadequate, because for these purposes there is a need for imposing different penalties to different types of errors. This idea leads to a further extension of the binary and multiclass machine learning paradigms, denoted with the term cost-sensitivity. Cost-sensitive algorithms can also be subdivided in different categories. Ordinal regression is a special case with the additional constraint of having an order on the classes. In this algorithm large errors with respect to the ranking (i.e. predicting $\mathcal{C}_1$ as $\mathcal{C}_3$ or vice versa) are more penalized than smaller errors (e.g. predicting $\mathcal{C}_1$ as $\mathcal{C}_2$). It is the underlying ranking function that guarantees this type of cost-sensitivity.

### 2.1 General definition of the circular ordinal model

Let us take the ordinal regression approach as a starting point to build a circular variant of the model. The discriminant function for ordinal regression looks like (1) and only considers one vector $\mathbf{w}$ instead of multiple vectors as in the regular multiclass models. Because this single function needs to learn a certain ranking on the given objects, the corresponding classification model is represented as,

$$h(\mathbf{x}) = \begin{cases} \mathcal{C}_1, \text{ if } f(\mathbf{x}) \leq b_1\,, \\ \mathcal{C}_2, \text{ if } b_1 < f(\mathbf{x}) \leq b_2\,, \\ \ldots \\ \mathcal{C}_r, \text{ if } b_m < f(\mathbf{x}). \end{cases} \tag{2}$$

where $r$ is the number of classes. The discriminant function $f(\mathbf{x})$ can be interpreted as a projection of the object $\mathbf{x}$ on the real line in such a way that it optimizes an order-preserving loss function when that real line is divided into different regions by the bias terms $b_k$. Figure 1a tries to visualize the idea behind the ordinal regression problem for the linear case where $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. One can clearly see the order that is present in the solution, as given by (2).

In a similar way we will now give the discriminant function and classification model for circular ordinal regression. Here, we will need two vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ to define a circular ordered structure as shown in Figure 1b. Therefore, we redefine the function $f$ as,

$$f : \mathcal{X} \mapsto \Re^2 : \mathbf{x} \mapsto \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \end{pmatrix} \cdot \Phi(\mathbf{x}) + \mathbf{b}. \tag{3}$$

Based on this discrimination function we reformulate the classification model $h$ as follows,

$$h(\mathbf{x}) = \mathcal{C}_k \quad \text{if} \quad \mathbf{v}_k^T f(\mathbf{x}) \geq 0 \quad \text{and} \quad -\mathbf{v}_{(k \bmod r)+1}^T f(\mathbf{x}) \geq 0\,, \tag{4}$$

with

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad \mathbf{v}_k = \begin{pmatrix} \cos(\frac{2\pi(k-1)}{r} + \frac{\pi}{2}) \\ \sin(\frac{2\pi(k-1)}{r} + \frac{\pi}{2}) \end{pmatrix} \quad \text{and} \quad k \in \{1, \ldots, r\}.$$

As with the ordinal case, we can view the discriminant function $f(\mathbf{x})$ as a projection. However, now it concerns a projection on the two-dimensional plane, such that the object $\mathbf{x}$ lies in its respective region around the origin (corresponding to its real label $\mathcal{C}_k$) defined by the vectors $\mathbf{v}_k$ and $\mathbf{v}_{(k \bmod r)+1}$. These two-dimensional vectors $\mathbf{v}_k$ represent the normals of the lines (through the origin) that separate the different class regions (see Figure 2.1). For an object, to lie in such a region, it has to be located in between two such separating lines. This means an object has to lie on the positive side as defined by the first normal of its corresponding class and on the negative side as defined by the second normal. Note that vector $\mathbf{v}_{(k \bmod r)+1}$ not only occurs in the second constraint for objects of class $\mathcal{C}_k$, but also in the first constraint for objects of the next adjacent class $\mathcal{C}_{(k \bmod r)+1}$. For objects of the last class $\mathcal{C}_r$, the second constraint makes use of the normal vector $\mathbf{v}_1$.

### 2.2 Primal cSVM formulation for four classes

Consider a data set of labeled objects $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $y_i \in \{1, \ldots, r\}$ is the class label of object $\mathbf{x}_i$. Now, we want to find a discriminant function $f$ that satisfies (4) for objects $\mathbf{x}_i$. To this end, we will use the idea of maximum separating hyperplanes as employed in SVMs. This leads to the following primal formulation,

$$\min_{\mathbf{w}_1,\mathbf{w}_2,b_1,b_2,\xi_i^{(1)},\xi_i^{(2)}} \frac{1}{2}(\|\mathbf{w}_1\|^2 + \|\mathbf{w}_2\|^2 + (b_1)^2 + (b_2)^2) + C\sum_{i=1}^{N} \xi_i^{(1)} + \xi_i^{(2)}\,, \tag{5}$$

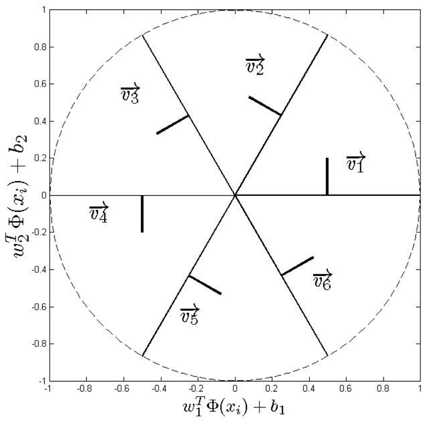


**Fig. 2.** An illustration of the meaning of the vectors $\mathbf{v}_k$ for the six-class cSVM. Each region in the circle is bounded by two lines, which are defined by their normal vectors $\mathbf{v}_k$. The orientation of these vectors is displayed as a black bar and does not represent the vector itself (as each vector should start in the origin). This was done for the purpose of visualization in order to show which vector and separating line belong together. By convention, we take the region for the first class as being defined by vectors $\mathbf{v}_1$ and $-\mathbf{v}_2$.

with constraints

$$\begin{aligned}
&(\mathbf{w}_1^T \varPhi(\mathbf{x}_i) + b_1)v_{y_i}^{(1)} + (\mathbf{w}_2^T \varPhi(\mathbf{x}_i) + b_2)v_{y_i}^{(2)} + \xi_i^{(1)} \geq 1\,, \\
&-(\mathbf{w}_1^T \varPhi(\mathbf{x}_i) + b_1)v_{y_i+1}^{(1)} - (\mathbf{w}_2^T \varPhi(\mathbf{x}_i) + b_2)v_{y_i+1}^{(2)} + \xi_i^{(2)} \geq 1\,, \\
&\xi_i^{(1)}, \xi_i^{(2)} \geq 0\,, \\
&\forall i \in \{1, \ldots, N\}\,,
\end{aligned}$$

where $v_{y_i+1}^{(1)}$ and $v_{y_i+1}^{(2)}$ represents the first and the second component of the two-dimensional vector $\mathbf{v}_{(y_i \bmod r)+1}$. For notational purposes we simply write $y_i + 1$ instead of $(y_i \bmod r) + 1$ as lower index, assuming $\mathbf{v}_{r+1}$ becomes $\mathbf{v}_1$. We also include slack variables $\xi_i^{(1)}$ and $\xi_i^{(2)}$, which represent the error corresponding to object $\mathbf{x}_i$. Although it is not the standard procedure, we also add the bias terms $b_1$ and $b_2$ in the objective function. In this way, we do not have to compute them explicitly, which greatly simplifies the implementation. Hsu *et. al.* [10] claimed that including the bias terms in the regularizer of the optimization problem does not result in substantially different solutions. Here, we also assume that this will not have a major impact on the solution of our method.

Before explaining the issue with the above given primal formulation (5), we briefly discuss the loss functions that can be used in traditional ordinal regression and circular ordinal regression. In traditional ordinal regression, a multitude of loss functions has been proposed in recent years (see [17] for an overview).

Despite the drawback of assuming a metric on the class labels, the mean-squared error (MSE) or the mean absolute error (MAE) are often considered as adequate loss functions. For example, in the four-class case, according to the MAE, an object of class one that is misclassified by ordinal regression in class four will receive the largest penalty. Formally, this loss on a dataset $D$ can be formulated as follows:

$$L(h, D) = \sum_{i=1}^{N} M_{y_i, h(\mathbf{x}_i)} \,, \tag{6}$$

where

$$M = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix} .$$

The rows $y_i \in \{1, 2, 3, 4\}$ represent the real labels, while the columns of $M$ represent the predicted labels. In case of circular ordinal regression, we use a slightly different matrix $M$ in the loss function. For example, for the circular mean absolute error and four classes, the matrix looks as follows:

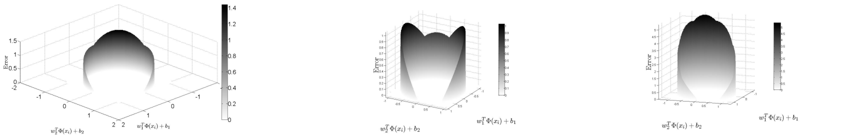
$$M = \begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix} . \tag{7}$$

Based on this circular loss function, we can see that objects of a certain class get the largest penalty when they are wrongly classified in the most opposite region, given the circular structure. For example, objects of the first class get the largest penalty when they are classified in the opposite quadrant or region corresponding to the third class. When the class labels are encoded in a specific way as binary relevance vectors (see Section 3), this circular loss function is equivalent with the Hamming loss. The Hamming loss is a popular loss function in multilabel classification problems and is defined in (9).

### 2.3 cSVM formulation for more than four classes

The error terms in the constraints of the primal formulation (5) should reflect this circular type of loss function. As we show in [8] this holds for four classes, but not for more than four (see Figure 3). This issue can be solved by adding additional constraints to the model. Instead of only using the normal vectors $\mathbf{v}_{y_i}$ and $\mathbf{v}_{y_i+1}$, delimiting the objects class region, we also include constraints based on the normal vectors that correspond to non adjacent boundaries (see Figure 2.1).

$$\min_{\mathbf{w}_1, \mathbf{w}_2, b_1, b_2, \xi_i^{(k)}} \frac{1}{2}(\|\mathbf{w}_1\|^2 + \|\mathbf{w}_2\|^2 + (b_1)^2 + (b_2)^2) + C \sum_{i=1}^{N} \sum_{k=1}^{r} \xi_i^{(k)} \,, \tag{8}$$



(a) Cost function induced by (5) for $r = 4$

(b) Cost function induced by (5) for $r = 8$

(c) Cost function induced by (8) for $r = 8$

**Fig. 3.** illustrates the problem with model (5) for more than four classes. Each figure displays the total error associated with an object of class one when projected on some point of the circle (represented by the $x$ and $y$ coordinate) according to (3). The left figure shows the loss function induced by the constraints of (5) for $r = 4$. One can clearly see that objects of class one get the largest penalty when classified in the most opposite region corresponding to objects of the third class. However, when we visualize the same loss function for eight classes in the center figure, we can not conclude the same. Apparently, in contrast to what we desire, the cost drops in the most opposite region. The loss function associated with formulation (8) solves this issue as one can see in the right figure.

with constraints

$$\begin{aligned}
&(\mathbf{w}_1^T \Phi(\mathbf{x}_i) + b_1)v_{k,y_i}^{(1)} + (\mathbf{w}_2^T \Phi(\mathbf{x}_i) + b_2)v_{k,y_i}^{(2)} + \xi_i^{(k)} \geq 1\,, \\
&\xi_i^{(k)} \geq 0\,, \\
&\forall i \in \{1, \ldots, N\}\,, \quad \forall k \in \{1, \ldots, r\}\,,
\end{aligned}$$

where

$$\mathbf{v}_{k,y_i} = s(k, y_i)\mathbf{v}_k\,,$$

and

$$s(k, y_i) = \begin{cases} -1 & \text{if} \quad k \in \{(y_i + 1) \bmod r, \ldots, (y_i + \frac{r}{2}) \bmod r\} \\ 1 & \text{otherwise.} \end{cases}$$

Now, each object is subject to $r$ constraints (indexed by $k$) instead of two in the previous model. The sign of the normal vectors $\mathbf{v}_k$ in each of the constraints now depends on the the label of the object.

The dual becomes

$$\max_{\alpha} -\frac{1}{2} \sum_{i,j,k,l} \alpha_i^{(k)} \alpha_j^{(l)} (K_{i,j}^{(k,l)} + \bar{K}_{i,j}^{(k,l)}) + \sum_{i,k} \alpha_i^{(k)}$$

subject to

$$\begin{aligned}
\alpha_i^{(k)} &\geq 0\,, \\
\forall i = 1 \ldots N, \forall k &= 1 \ldots r\,,
\end{aligned}$$

with

$$K_{i,j}^{(k,l)} = v_{k,y_i}^{(1)} v_{l,y_j}^{(1)} (\Phi^T(x_i)\Phi(x_j) + 1)$$
$$\bar{K}_{i,j}^{(k,l)} = v_{k,y_i}^{(2)} v_{l,y_j}^{(2)} (\Phi^T(x_i)\Phi(x_j) + 1).$$

This is a standard quadratic program with linear inequality constraints which can be solved easily with most convex optimizaton software packages. The matrix in the quadratic term of the objective function grows quadratically according to $rN$.

For a new point $\mathbf{x}$ we can make a two-dimensional projection as follows

$$f(\mathbf{x}) = \begin{pmatrix} \sum_{i=1}^{N} \sum_{k=1}^{r} \alpha_i^{(k)} v_{k,y_i}^{(1)} (\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) + 1) \\ \sum_{i=1}^{N} \sum_{k=1}^{r} \alpha_i^{(k)} v_{k,y_i}^{(2)} (\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}) + 1) \end{pmatrix}.$$

Using (4) we can then predict its class.

## 3 Circular ordinal regression as multilabel classification

Circular ordinal regression can also be written as a multilabel classification problem, by transforming the original labels to vectors of binary labels. One of the most simple methods for multilabel classification is binary relevance, where each classifier predicts exactly one label component [5]. The output of such a multilabel classifier $\mathbf{h}$ is a vector

$$\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_{\frac{r}{2}}(\mathbf{x})).$$

In our conversion of circular ordinal regression, we only need $\frac{r}{2}$ binary classifiers for reasons that will become clear in a minute. Our goal is now to re-encode the original labels as binary vectors $\mathbf{y} = (y^{(1)}, \ldots, y^{(\frac{r}{2})})$ (where $y^{(k)} = -1$ or $y^{(k)} = 1$) in order to impose the circular ranking. In other words, we need to transform the original labels in a way that large errors (as defined by equation (6) with $M$ given by (7)) are penalized most. Off course, this can only be done with respect to a certain loss function. To this end, we consider the Hamming loss, a popular loss function in multilabel classification, defined as

$$L_H(\mathbf{y}, \mathbf{h}(\mathbf{x})) = \sum_{i=1}^{\frac{r}{2}} [\![ y^{(i)} \neq h_i(\mathbf{x}) ]\!]. \tag{9}$$

Now we have all information necessary to construct a proper code for each class label. Let us take the example of six classes. Then,

$$\mathcal{C}_1 = (-1,-1,-1), \quad \mathcal{C}_2 = (-1,-1,1), \quad \mathcal{C}_3 = (-1,1,1),$$
$$\mathcal{C}_4 = (1,1,1), \quad \mathcal{C}_5 = (1,1,-1), \quad \mathcal{C}_6 = (1,-1,-1)\,,$$

would be a good code when the classes are circular ordered, *e.g.* $\mathcal{C}_2$ and $\mathcal{C}_6$ are the classes adjacent to $\mathcal{C}_1$. For this coding, one can easily notice that the circular mean absolute error coincides with the Hamming loss. A general rule for constructing the circular binary labels is given by

$$y_i^{(j)} = \begin{cases} 1 & \text{if } \mathbf{x}_i \in \{\mathcal{C}_{\frac{r}{2}+2-j}, \ldots, \mathcal{C}_{r+1-j}\} \\ -1 & \text{otherwise} \end{cases}$$

Thus, in general, for $r$ classes we need to construct $\frac{r}{2}$ binary classifiers to make the Hamming identical to the circular mean absolute error. This is the main difference with the method presented before (see Section 2), where we only construct two models independently of the number of classes. However, for $r = 4$ both methods are exactly the same. This can be seen by first writing the objective function for the multilabel approach as a single optimization problem and then comparing it with (5):

$$\min_{\mathbf{w}_1,\mathbf{w}_2,b_1,b_2,\xi_i^{(1)},\xi_i^{(2)}} \frac{1}{2}(\|\mathbf{w}_1\|^2 + \|\mathbf{w}_2\|^2 + (b_1)^2 + (b_2)^2) + C\sum_{i=1}^{N} \xi_i^{(1)} + \xi_i^{(2)}\,, \quad (10)$$

with constraints

$$\begin{aligned} & y_i^{(1)}(\mathbf{w}_1^T \Phi(\mathbf{x}_i) + b_1) \geq 1 - \xi_i^{(1)} \\ & y_i^{(2)}(\mathbf{w}_2^T \Phi(\mathbf{x}_i) + b_2) \geq 1 - \xi_i^{(2)} \\ & \xi_i^{(1)}, \quad \xi_i^{(2)} \geq 0 \\ & \forall i \in \{1, \ldots, N\}. \end{aligned}$$

The optimization problem in (5) has the same objective function, with the following equivalent constraints,

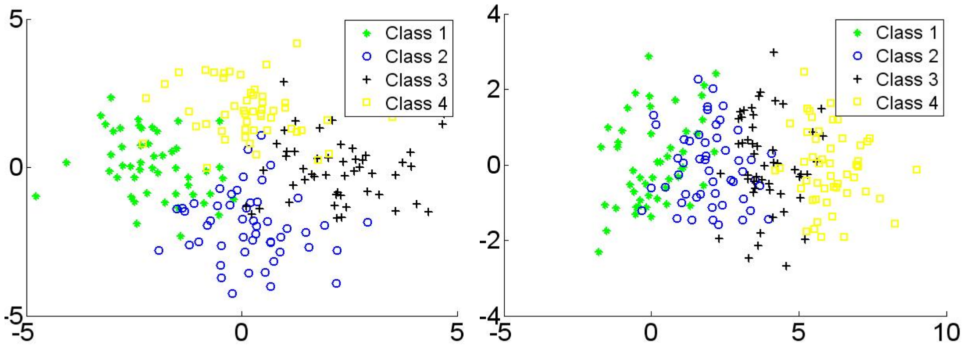
$$\begin{aligned} \mathbf{w}_2^T \Phi(\mathbf{x}_i) + b_2 + \xi_i^{(2)} &\geq 1, & \forall x_i \in \mathcal{C}_1 \cup \mathcal{C}_2 \\ \mathbf{w}_1^T \Phi(\mathbf{x}_i) + b_1 + \xi_i^{(1)} &\geq 1, & \forall x_i \in \mathcal{C}_1 \cup \mathcal{C}_4 \\ -\mathbf{w}_1^T \Phi(\mathbf{x}_i) - b_1 + \xi_i^{(1)} &\geq 1, & \forall x_i \in \mathcal{C}_2 \cup \mathcal{C}_3 \\ -\mathbf{w}_2^T \Phi(\mathbf{x}_i) - b_2 + \xi_i^{(2)} &\geq 1, & \forall x_i \in \mathcal{C}_3 \cup \mathcal{C}_4 \\ \xi_i^{(1)}, \xi_i^{(2)} &\geq 0. \\ \forall i \in \{1, \ldots, N\}. \end{aligned}$$

where we choose the vectors $\mathbf{v}$ as,

$$\mathbf{v}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \mathbf{v}_3 = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \mathbf{v}_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Using the following code for relabeling the original labels as binary vectors,

$$\begin{aligned} & \mathcal{C}_1 = (1, 1), \quad \mathcal{C}_2 = (-1, 1), \\ & \mathcal{C}_3 = (-1, -1), \mathcal{C}_4 = (1, -1)\,, \end{aligned}$$



(a) Toy example A (b) Toy example B

**Fig. 4.** shows the first two dimensions of the training set for toy examples A and B. Toy example A contains four classes for which the class conditional means lie on a circle with radius two. For toy example B the class conditional means lie along the first dimension.

we can immediately see the equivalence between both constraint sets. Both methods project the data in the two-dimensional plane so that all objects lie in their respective quadrants. However, for more than four classes, the binary relevance multilabel approach uses more hyperplanes to construct the discrimination model and thus projects the data into a higher dimensional space compared to the first approach.

Concerning the multilabel approach, a subtle problem arises during prediction in case of more than four classes. For example in the six class case, it could be possible that the binary relevance classifier makes the following prediction $(-1, 1, -1)$ which does not correspond to any of the six original class labels. However, this can be solved easily by using the continuous outputs of the binary classifiers and computing the inner product with all allowed binary vectors, choosing the one with the highest value as the final class label.

## 4 Experiments

We consider two toy examples A and B to check the operation of the circular ordinal model. We also apply the model to real-life data of the brain-computer interface (BCI) competition [1].

### 4.1 Description

**Toy examples** For toy example A the cluster centers of each class lie on a circle embedded in a higher dimensional space. Simulations are done with both ten and 200 dimensional data for four classes with 50 objects in each class. The first two dimensions have a Gaussian distribution with the mean lying on the boundary of a circle with radius two. The remaining dimensions are completely random, displaying no discriminative information. The means of each class are spread uniformly across the circle so that the overlap between class-conditional distributions is equal. An example set for four classes is shown in Figure 4a. The test set is constructed in a similar way as the training set.

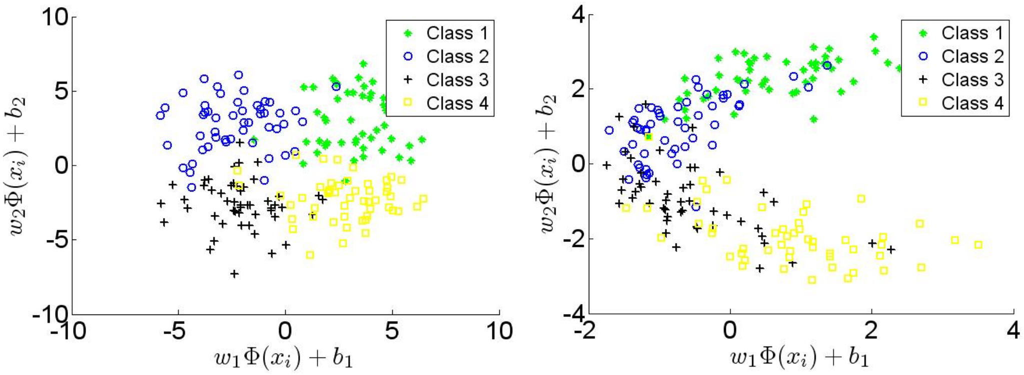
In toy example B, we do a similar experiment where the cluster centers of each class now lie on a straight line across the first dimension. The difference between the class conditional means is two so that there is an equal overlap between the class distributions. Here, we also consider four classes with 50 objects each, again once with ten dimensions and once with 200 dimensions. An example set is shown in Figure 4b.

**BCI data** Currently, BCI is a hot topic, bridging a gap between computer science and neuroscience. It can give people with certain disabilities an alternative communication pathway or can be used in the context of neurofeedback. There are different types of BCI, depending on the imaging technology being used. One of the most popular imaging techniques for BCI is electroencephalography (EEG). EEG measures the potential difference on the scalp of a subject over a number of channels. Different signal processing algorithms and machine learning techniques can then be used to extract information about the user's intention from the EEG. Periodically, a contest is held to test different algorithms on several problems within the BCI domain. The third BCI competition [1] contains such a data set IIIa of EEG samples belonging to four different conditions. The four conditions represent the imagination of left hand, right hand, foot and tongue movement recorded across 60 electrodes. Looking at the possible application of this kind of data, we can clearly see a circular order present on the labels. For example imagine an application where the user wants to control a computer cursor or a wheel chair by imagining one of the above movements. Why is this circular? For example, if a user wants to steer a wheel chair straight ahead, but slightly deviates to the right or left, this is a less severe error than completely turn around and go back.

Before taking a look at the results, we briefly describe the feature extraction algorithm. Firstly, the signal is filtered between different frequency ranges (each filter bank having a range of 4hz) and a popular spatial filter is applied to each of them, namely common spatial patterns (CSP) [13]. For the multiclass CSP algorithm, we use the one-versus-all approach. In this way, each spatial filter corresponds to one of the classes and for the most discriminative spatial filters the most important filter banks are chosen based on the Fisher ratio. Next, the temporal and spatial filtered EEG signal is divided in epochs (the middle two seconds of each trial). For each of these epochs we calculate the variance. The variances are then used as train objects for the classifier.

### 4.2 Results

**Toy examples** In Table 1 and 2 the results are shown for toy example A and B. In order to compare the methods we use two different performance measures.



(a) Toy example A (b) Toy example B

**Fig. 5.** Predictions according to (3) for cSVM.

The first and most common one is the accuracy. Secondly, we give results for the Hamming loss as defined by (6) with a matrix $M$ equal to (7). For toy examples A and B, we generate 30 different train and test sets and compute the mean for the two above mentioned performance measures. We also present the results for standard multiclass SVM, both for one-versus-one and one-versus-all.

**Table 1.** Results for toy example A.

| | Dimension | BR | cSVM | 1-vs-all | 1-vs-1 |
|---|---|---|---|---|---|
| Hamming | 10 | 0.172 | 0.169 | 0.174 | 0.180 |
| | 200 | 0.276 | 0.279 | 0.341 | 0.324 |
| Accuracy | 10 | 83.7 | 83.9 | 83.5 | 83.0 |
| | 200 | 74.2 | 73.8 | 68.2 | 69.9 |

**Table 2.** Results for toy example B.

| | Dimension | BR | cSVM | 1-vs-all | 1-vs-1 |
|---|---|---|---|---|---|
| Hamming | 10 | 0.289 | 0.286 | 0.263 | 0.269 |
| | 200 | 0.524 | 0.524 | 0.423 | 0.410 |
| Accuracy | 10 | 71.5 | 71.7 | 73.9 | 73.4 |
| | 200 | 52.0 | 51.3 | 60.3 | 61.1 |

The default Gaussian kernel is used to train the models. Thus, we need to determine the parameters $C$ and $\gamma$ through cross-validation. After cross-validation, we apply the learned model to the generated test set in order to get two-dimensional predictions, as shown in Figure 5.

For the first toy example A, Table 1 shows that the circular models (binary relevance and cSVM) perform better than the classical SVM approaches (one-versus-one and one-versus-all) both in terms of Hamming loss and accuracy. This difference becomes significant under the Wilcoxon signed rank test when the number of features increases. While varying the overlap between class-conditional distributions, we also observed that the difference between both the circular and classical methods decreased, when the overlap decreased, until both type of methods performed equal on both measures. This supports the findings of [5–7], which state that the multilabel risk minimizers of the Hamming loss and the subset 0/1 loss (this is accuracy in our case) coincide when the joint mode equals the marginal modes (i.e., the modes computed for each label individually).

Intuitively, toy example B seems a lot more challenging for the circular models, because for the first label it has to build a representation for the inner two distributions and for the outer two, while classic multiclass one-versus-one or one-versus-all SVMs can model each distribution separately. Indeed, Table 2 shows that toy example B is a lot harder. The results are similar to the previous example, except it is now in favor of the classical SVM models. According to [5–7] and due to the conditonal dependency of the labels, it is not unlikely that the standard multiclass SVMs perform better in terms of accuracy. However, at first, it is unexpected to see that these standard models also outperform the circular models in terms of the Hamming loss, while it suffices to model the marginal distributions. On the other hand, the data of toy example B is not structured in a circular way, which could explain why the circular models also perfom worse in terms of the Hamming loss.

In both toy examples, cSVM and the binary relevance approach compare well to each other, which confirms the similarity between them. The small difference between them is probably due to different implementations. A similar small difference is observed for the standard multiclass approaches, which perform almost identical, with a small advantage for the one-versus-one method when the dimensionality increases.

**Table 3.** Results for the BCI data. We also give the percentage of large errors: objects that are penalized with the largest error according to matrix (7).

| | Subject | BR | cSVM | 1-vs-all |
|---|---|---|---|---|
| Hamming | k3b | 0.25 | 0.31 | 0.24 |
| | k6b | 0.57 | 0.57 | 0.52 |
| | l1b | 0.55 | 0.54 | 0.53 |
| Accuracy | k3b | 77 | 71 | 79 |
| | k6b | 55 | 53 | 59 |
| | l1b | 53 | 53 | 58 |
| Large errors (%) | k3b | 1 | 1 | 1 |
| | k6b | 7 | 10 | 11 |
| | l1b | 5 | 5 | 5 |

**BCI data** The results for the BCI data are given in Table 3 and computed in a special way. For computing the performance measures, we employ the method proposed in [14]. Each measure is calculated and averaged at each sample over all trials of the test set. In this way, we become a time course of the respective measure. For the accuracy we then use the maximum occurring value as the final measure, while for the number of large errors and the Hamming loss we use the minimum. In terms of accuracy and Hamming loss the classical SVM approach seems to perform best, only for the percentage of large errors the circular models have a small advantage. We do not have an explanation for the relatively large difference between the binary relevance and the cSVM method on subject *k3b*.

## 5 Discussion

We presented two different but related methods for circular ordinal regression. The first approach (cSVM) only used two hyperplanes independently of the number of classes and was shown to be equivalent with the second approach (binary relevance multilabel classifier) in case of four classes. To this end, we showed that the circular ordinal regression problem can be simply cast as a multilabel classification problem by encoding the original labels in a specific way.

Initial experimental results more or less confirmed what could be expected beforehand; the two presented methods clearly outperformed multi-class classifiers on a benchmark problem where the underlying model exhibits a circular structure. Importantly, the circular models improve in such a situation for Hamming loss and accuracy, because the minimizers of these loss functions coincide. On the other hand, the multi-class classifiers outperformed the circular models when the underlying data generation process deviated from a circular structure, both in terms of Hamming loss and accuracy. Furthermore, the difference between optimizing different loss functions was a bit more visible in the BCI application. Here the circular models were capable of reducing the number of large errors, thereby leading to a decrease in accuracy. However, since the number of large errors was very low in this dataset, the BCI application should be seen as a motivating example rather than a statistically relevant benchmark problem.

From a different perspective, the obtained results provide additional insights concerning the recent discussion of risk minimization in multilabel classification. Contrary to the theoretical results of [5–7], the binary relevance approach did not always perform better in terms of the Hamming loss. We suspect that this is caused by the specific distribution of the data and the capacity of the models. However, since RBF kernels were used, the different models should have a similar capacity. Further investigation of this unexpected observation is necessary. In addition, further experiments for the case of more than four circular ordinal classes have to be set up as well.

## References

1. BCI Competition III. Tech. rep., IDA Fraunhofer first (2005), Available at http://www.bbci.de/competition/iii/
2. Caruana, R., Baluja, S., Mitchell, T.: Using the future to sort out the present: Rankprop and multitask learning for medical risk evaluation. In: Advances in neural information processing systems 8. vol. 8, pp. 959–965. Denver (1996)
3. Chu, W., Keerthi, S.: Support vector ordinal regression. Neural computation 19(3), 792–815 (2007)
4. Crammer, K., Singer, Y.: Pranking with ranking. In: Advances in neural information processing systems 14. vol. 14, pp. 641–647. Vancouver, Canada (2002)
5. Dembczyński, K., Cheng, W., Hüllermeier., E.: Bayes optimal multilabel classification via probabilistic classifier chains. In: Proceedings of the International Conference on Machine Learning (2010), to appear
6. Dembczyński, K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence in multilabel classification. In: Proceedings of the ICML Workshop on Learning from Multi-label data, Haifa, Israel. pp. 5–13 (2010)
7. Dembczyński, K., Waegeman, W., Cheng, W., Hüllermeier, E.: Regret analysis for performance metrics in multilabel classification: the cae of hamming and subset zero-one loss. In: Proceedings of the European Conference on Machine Learning, Barcelona, Spain (2010), to appear
8. Devlaminck, D., Waegeman, W., Bauwens, B., Wyns, B., Otte, G., Boullart, B., Santens, P.: Directional predictions for 4-class bci data. In: Proceedings of the 18th European Symposium On Artificial Neural Networks, Bruges, Belgium. pp. 219–224 (2010)
9. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Proceedings of the 12th European Conference on Machine Learning. pp. 145–156. Freiburg, Germany (2001)
10. Hsu, C., Lin, C.: A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 13(2), 415–425 (Mar 2002)
11. Jianlin, C., Zheng, W., Pollastri, G.: A neural network approach to ordinal regression. In: 2008 IEEE International Joint Conference on Neural Networks. pp. 1279–84 (2008)
12. Ling, L., Hsuan-Tien, L.: Ordinal regression by extended binary classification. In: Advances in Neural Information Processing Systems 19. pp. 865–872. Vancouver, Canada (2007)
13. Müller-Gerking, J., Pfurtscheller, G., Flyvbjerg, H.: Designing optimal spatial filters for single-trial eeg classification in a movement task. Clinical Neurophysiology 110(5), 787–798 (1999)
14. Schlögl, A.: Biosig - an open source software library for biomedical signal processing. (2003-2004), http://biosig.sf.net
15. Shashua, A., Levin, A.: Ranking with large margin principle: two approaches. In: Advances in Neural Information Processing Systems 15. pp. 937–944. Vancouver, Canada (2003)
16. Waegeman, W., De Baets, B.: On the ERA ranking representability of multi-class classifiers. Artificial Intelligence (2010), to appear
17. Waegeman, W., De Baets, B., Boullart, L.: ROC analysis in ordinal regression learning. Pattern Recognition Letters 29, 1–9 (2008)