IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 14, 2020, accepted August 29, 2020, date of publication September 3, 2020, date of current version September 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021596

Ordinal Classification: Working Definition and Detection of Ordinal Structures

PETER BELLMANN AND FRIEDHELM SCHWENKER, (Member, IEEE)
Institute of Neural Information Processing, Ulm University, 89081 Ulm, Germany

Corresponding author: Peter Bellmann (peter.bellmann@uni-ulm.de)

The work of Peter Bellmann and Friedhelm Schwenker is supported by the Project Multimodal Recognition of Affect over the course of a tutorial learning experiment funded by the German Research Foundation (DFG) under Grant SCHW623/7-1.

ABSTRACT Ordinal classification (OC) is an important niche of supervised pattern recognition, in which the classes constitute an ordinal structure. In general, the ordinal structure can be identified, either according to the natural occurrence of the current task (e.g. healthy - mild condition - moderate condition - severe condition), or by extracting expert knowledge. However, we assume that many multi-class classification tasks might have a hidden ordinal structure, which, once identified, can facilitate and hence leverage the classification process. Therefore, we propose a working definition for OC tasks, which is based on the decision boundaries of standard binary Support Vector Machines. Moreover, resulting from our proposed definition, we introduce a simple algorithm for the detection of ordinal structures. Our proposed definition is easy to interpret and reflects an intuitive understanding of ordinal structures. Another main advantage is that our proposed definition is easy to apply. Therefore, there is no more dependence on expert knowledge for the identification of (non-intuitive) ordinal class structures. In the current study, we include ten benchmark data sets from the field of OC to experimentally evaluate and hence to confirm the validity of our proposed definition. Additionally, we analyse our proposed definition based on a small set of traditionally non-ordinal multi-class classification tasks. Furthermore, we provide an analysis of the computational cost of our proposed detection algorithm, and discuss the limitations of our proposed working definition.

INDEX TERMS Detection of ordinal class structures, ordinal classification, support vector machines.

I. INTRODUCTION

Ordinal classification (OC), sometimes also referred to as ordinal regression, represents a special category of supervised pattern recognition. In OC tasks, it is assumed that the classes are arranged according to a natural order, e.g. *short* $\prec$ *normal* $\prec$ *long*. Moreover, it is assumed that the natural order is reflected in the feature space of the data. Thus, the presence of expert knowledge about the ordinal structure of a given data set constitutes an important additional information, which can be used to improve the classification accuracy. Different studies, such as the one proposed by Hühn and Hüllermeier [19] confirm that ordinal structures are indeed useful in classifier learning.

Ordinal regression and classification (ORC) are two very interesting research fields, which have been analysed by many researches for decades, e.g. [25], and which can easily fill books, e.g. [2]. ORC tasks still constitute up-to-date topics. This can be easily observed by the fact that, a few years ago, different researchers started to adapt deep neural models to ordinal data, e.g. [8], [24], [26]. For one of the most recent surveys on ORC, we refer the reader to a study proposed by Gutiérrez *et al.* [18].

Over the years, existing classification approaches have been modified for a better adaptation to the specific ORC tasks. Especially, Support Vector Machines (SVMs) seem to be popular tools for such modification purposes, e.g. [9], [10]. As an example, Cardoso *et al.* [6] used Support Vector Machines in an ordinal setting for an objective aesthetic evaluation of breast cancer conservative treatment.

In fact, health or psychological applications, such as breast cancer and pain recognition analyses, make ORC tasks so important. Thereby, the ordinal classes usually represent different stages of diseases or levels of pain, e.g. *no pain* $\prec$ *low pain* $\prec$ *intermediate pain* $\prec$ *strong pain*, e.g. [4], [28]. Even different classification performance metrics have been proposed to appropriately fit to the specific ORC tasks, e.g. [7], [11]. The need for ORC-specific classification performance measures is justified by the different fields of application, such as the aforementioned example of pain

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

intensity recognition. Intuitively, it is safe to assume that there is a substantial difference between misclassifying the no-pain level as the high-pain level, and misclassifying the no-pain level as the low-pain level.

In general, the information about a data set's ordinal structure is provided intuitively by the natural characteristics of the corresponding classification task. As mentioned above, this is especially the case for medical applications. But what if the ordinal structure is *hidden*, i.e. what if the task is not obviously ordinal, in a natural way? Then, we need an expert who provides us with the information about the ordinal structure.

It is to expect that we all understand how to interpret the fact that a given data set constitutes an OC task. However, we believe, currently there is still a need for task-specific experts, because there is still a lack of a common definition for OC tasks, which is not based on the natural occurrence of the corresponding classes. Therefore, in the current study, we introduce a *working definition* for OC tasks, which we denote as *SVM based ordinal* (SVM-ordinal), and which fulfils the following two properties. First, our proposed working definition is easy to interpret. Second, our working definition can be applied easily. Especially, this means that, based on our proposed definition, one is able to identify OC tasks, without any kind of additional expert knowledge.

The remainder of this work is organised as follows. In Section II, we provide some related work and the motivation for our proposed working definition. We introduce our proposed working definition in Section III, including an interpretation and discussion, as well as an additional theory-based consequence for 3-class classification tasks. In Section IV, we introduce a simple method for the detection of ordinal structures. The ordinal classification benchmark data sets, which are evaluated to confirm the validity of our proposed working definition, are shortly described in Section V, followed by the presentation and discussion of our results. Moreover, in Section VI, we include a short analysis based on common multi-class tasks, which are non-ordinal in the traditional sense. We discuss the operational complexity of our presented detection algorithm, as well as the limitations of our proposed definition, in Section VII. Finally, in Section VIII, we conclude this study.

II. PRELIMINARIES, RELATED WORK AND MOTIVATION

In the current section, we shortly describe the functionality of binary Support Vector Machines. Subsequently, we briefly summarise the concept of cascaded classification systems, which are popular tools in OC tasks. Finally, we introduce a couple of studies on the detection of ordinal class structures, which provided the inspiration for our current work.

A. SUPPORT VECTOR MACHINES FOR BINARY AND MULTI-CLASS TASKS

Abe introduced the Support Vector Machine (SVM) [1]. An SVM is a binary classifier that combines two objectives. First, an SVM finds a hyperplane, which separates the two classes from each other. Second, in addition, the SVM maximises the *margin*. The margin is defined as the space surrounding the hyperplane, which does not include any observations. For the case of inseparable classes, the two objectives remain the same. However, an additional cost term is included, which penalises the width of the margin, for every data sample, which is located on the wrong side of the hyperplane. Initially implemented for binary classification tasks, the SVM algorithm can be easily extended to solve multi-class classification tasks by applying one of the existing divide-and-conquer approaches (DCAs), such as the *error correcting output codes* (ECOC) [12]. Popular ECOC models include the *one-versus-one* approach, as well as the *one-versus-all* approach. For a current data sample, the outputs of the corresponding binary classifiers are combined by an intelligent aggregation rule, which leads to the final decision. Many different types of binary subtask models can be defined, such as the so-called *ternary* ECOC classifiers proposed by Escalera *et al*. [14], [15]. For further DCAs, we refer the reader to the two studies proposed by Allwein *et al*. [3], and Tax and Duin [27].

B. CASCADED CLASSIFICATION ARCHITECTURES

The above mentioned ECOC methods work in parallel manner. Each classifier's output is used and combined by a corresponding combination function to obtain the final decision. In the cascaded classification approach, e.g. [17], the labelling of a data sample is proceeded sequentially. Thereby, the classification models are arranged as a kind of classification chain. Each of the chain members can choose between two options. Either the current classifier decides the label of the corresponding input sample, or the current classifier decides to *move* the current sample to the next classifier. Thus, cascaded classification architectures (CCAs) constitute ensemble selection techniques [21], in which the output specific to solely one ensemble member is defined as the final ensemble decision. Let $c \in \mathbb{N}$, $c > 2$, be the number of classes, with the class labels $\omega_1 \prec \ldots \prec \omega_c$. In general, CCAs consist of a chain of $c - 1$ specific classification models. The classification models define the type of the CCA. For example, in the *lower-versus-higher* (*higher-versus-lower*) approach, the classification model on position i is trained to separate the classes $\{\omega_1, \ldots, \omega_i\}$ ($\{\omega_c, \ldots, \omega_i\}$) from the classes $\{\omega_{i+1}, \ldots, \omega_c\}$ ($\{\omega_{i-1}, \ldots, \omega_1\}$). Whereas in the *pairwise* approach, the classification model on position i is trained to separate the class ω_i from the class ω_{i+1}. For all CCA approaches it holds, if the output of classifier i corresponds to class ω_i, then this output is taken as the architecture's final decision. Otherwise, the input sample is moved to the classifier on position $i + 1$.

C. RECENT STUDIES ON THE DETECTION OF ORDINAL STRUCTURES

Lattke *et al*. [22] introduced a method for detecting ordinal structures based on the evaluations of different CCAs. The main idea of their approach can be summarised as follows.

The accuracy performance of a CCA is sensitive to the order, and hence to the binary classification subtasks, of its chain members. The authors propose to consider all possible order permutations of the classes. For each permutation, one has to apply a k-fold cross validation, $k \in \mathbb{N}, k > 1$, in combination with the corresponding CCA. Assume that we have a labelled data set with the inherent order $\omega_1 \prec \omega_2 \prec \omega_3$. Then it is assumed that the CCA, which was trained according to the label set $\{\omega_1, \omega_2, \omega_3\}$ leads to significantly better classification results than, for example, the CCAs that were trained according to the label sets $\{\omega_2, \omega_1, \omega_3\}$ or $\{\omega_3, \omega_1, \omega_2\}$. The outcomes presented by Lattke *et al.* [22] indicate that one can detect ordinal class structures by applying CCAs, in combination with linear SVM models.

Lausser *et al.* [23] showed that this specific approach can be used to assess phenotype order in molecular data. Such kind of data represents one of the many interesting tasks, in which we usually do not have any a priori information about the corresponding structure.

D. MOTIVATION

Lattke *et al.* [22] and Lausser *et al.* [23] proposed to implement pairwise cascaded SVM models to detect ordinal structures. Based on the outcomes of those studies, we believe that one can also approach the whole idea the other way around. Therefore, if one can identify ordinal structures in combination with SVM classifiers, then one can also provide a common definition for ordinal classification tasks, which is based on SVMs. Thus, in the following section, we will introduce a working definition for ordinal classification tasks, which is based on the corresponding binary decision boundaries, i.e. hyperplanes, which are provided by standard SVMs.

III. WORKING DEFINITION FOR ORDINAL CLASSIFICATION TASKS

In the current section, we first provide all necessary notations, followed by the formulation of our SVM-based definition for ordinal classification tasks. Subsequently, we provide a brief interpretation and discussion on our proposed definition. We complete this section by presenting a theorem on the ordinal structure of common 3-class classification tasks.

Note that throughout this whole study, each SVM will denote a linear SVM, i.e. a Support Vector Machine with linear kernel (For the choice of the linear kernel, see Sec. II-C).

A. FORMALISATION

Let $X \subset \mathbb{R}^d, d \in \mathbb{N}$, be a d-dimensional labelled data set. By $\Omega = \{\omega_1, \dots, \omega_c\}, c \geq 3$, we denote the set of class labels. Moreover, by $l(x)$, we denote the true label of $x \in X$. For all $i, j = 1, \dots, c$, with $i \neq j$, we define the subset of X, which includes all samples from the two classes ω_i and ω_j, as $X_{i,j}$, i.e.

$$X_{i,j} := \{x \in X : l(x) = \omega_i \vee l(x) = \omega_j\}.$$

By $SVM_{i,j}$, we denote a support vector machine, which is trained to learn the binary classification task specific to $X_{i,j}$, i.e. $SVM_{i,j} : \mathbb{R}^d \to \Omega_{i,j} := \{\omega_i, \omega_j\} \subset \Omega, \forall i, j = 1, \dots, c$, $i \neq j$. Moreover, by $acc_{i,j}$, we denote the resubstitution accuracy of classifier $SVM_{i,j}$, i.e.

$$acc_{i,j} := \frac{|\{x \in X_{i,j} : SVM_{i,j}(x) = l(x)\}|}{|X_{i,j}|}, \tag{1}$$

whereby $|\cdot|$ denotes the number of instances in a set.

Note that in our illustrations, we will use the corresponding percentage values, i.e. $acc_{i,j} \times 100$, for better readability.

B. WORKING DEFINITION

Let $\mathcal{T}^c$ be the set of all permutations of the set $\{1, \dots, c\}$, i.e. $\mathcal{T}^c \ni \tau : \{1, \dots, c\} \to \{1, \dots, c\}$, and τ is a bijective function. Further, let $\mathcal{A}_\Omega \in \mathbb{R}^{c \times c}$ be the *pairwise accuracy matrix* (PAM), with elements $\mathcal{A}_\Omega = (a_{i,j})_{i,j=1}^c$, which we define as follows,

$$a_{i,j} := \begin{cases} acc_{i,j}, & \text{if } i \neq j, \\ 0, & \text{if } i = j. \end{cases} \tag{2}$$

Note that, by definition, the matrix $\mathcal{A}_\Omega$ is symmetric. The entry of matrix $\mathcal{A}_\Omega$, which is located in row i and column j, i.e. $a_{i,j}$, denotes the resubstitution accuracy of the support vector machine $SVM_{i,j}$, which is trained on $X_{i,j}$.

By $\mathcal{A}_\Omega^{(\tau)}$, we denote the PAM, whose elements are defined as $a_{i,j}^{(\tau)} := acc_{\tau(i),\tau(j)}$, for all $i \neq j$, and $a_{i,i}^{(\tau)} := 0$, for all $i = 1, \dots, c$. Note that, by definition, the matrix $\mathcal{A}_\Omega^{(\tau)}$ is symmetric as well. Therefore, with *id* denoting the *identity* permutation of the set $\mathcal{T}^c$, i.e. $id : \{1, \dots, c\} \mapsto \{1, \dots, c\}$, it holds, $\mathcal{A}_\Omega = \mathcal{A}_\Omega^{(id)}$. Moreover, once the elements $a_{i,j}$ of $\mathcal{A}_\Omega$ are known, one can determine $\mathcal{A}_\Omega^{(\tau)}$, for any $\tau \in \mathcal{T}^c$, by simply applying τ to $\mathcal{A}_\Omega$, i.e. $a_{i,j}^{(\tau)} = a_{\tau(i),\tau(j)}$. In addition, for each $\tau \in \mathcal{T}^c$, we define the *reverse* permutation of τ by $-\tau \in \mathcal{T}^c$, e.g. $-id : \{1, \dots, c\} \mapsto \{c, c-1, \dots, 1\}$.

Let us assume, that the labelled data set $X \subset \mathbb{R}^d$ constitutes an ordinal c-class classification task, with respect to the initial order of the label set Ω. Then, by our proposed definition, the structure of the corresponding PAM, $\mathcal{A}_\Omega = \mathcal{A}_\Omega^{(id)}$, has to fulfil the below defined properties, which can be depicted as follows,

$$\mathcal{A}_\Omega = \begin{pmatrix} 0 & \leq a_{1,2} & \leq & \cdots & \leq & a_{1,c} \\ a_{2,1} & \geq \ 0 & \leq & \cdots & \leq & a_{2,c} \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ a_{c-1,1} & \geq \ \cdots & \geq & 0 & \leq & a_{c-1,c} \\ a_{c,1} & \geq a_{c,2} & \geq & \cdots & \geq & 0 \end{pmatrix}. \tag{3}$$

Definition 1 (Working Definition for Ordinal Classification): Let $X \subset \mathbb{R}^d, d \in \mathbb{N}$, be a d-dimensional labelled data set with the set of class labels, $\Omega = \{\omega_1, \dots, \omega_c\}$, with $c \geq 3$.

Permutation $\tau \in \mathcal{T}^c$ represents an *ordinal class structure* if and only if $\forall i, j, k \in \{1, \dots, c\}$, the corresponding PAM $\mathcal{A}_\Omega^{(\tau)}$

fulfils the following properties,

$$\left.\begin{array}{rl} a_{i,j}^{(\tau)} \geq a_{i,k}^{(\tau)} & \forall j < k \leq i, \\ \wedge \quad a_{i,j}^{(\tau)} \leq a_{i,k}^{(\tau)} & \forall i \leq j < k. \end{array}\right\} \tag{4}$$

We define X as *SVM based ordinal (SVM-ordinal), with respect to* $\nu, -\nu \in \mathcal{T}^c$, if and only if

PAM $\mathcal{A}_\Omega^{(\tau)}$ fulfils Eq. (4) for $\tau \in \{\nu, -\nu\}$, and
PAM $\mathcal{A}_\Omega^{(\tau)}$ violates Eq. (4) $\forall \tau \in \mathcal{T}^c \backslash \{\nu, -\nu\}$.

Therefore, we shortly say that the labelled data set X constitutes an SVM-ordinal classification task, if there exist *exactly two* permutations $\tau, -\tau \in \mathcal{T}^c$, such that Equation (4) becomes true, for the corresponding PAMs $\mathcal{A}_\Omega^{(\tau)}$ and $\mathcal{A}_\Omega^{(-\tau)}$. Note that by definition, it directly follows, if $\tau \in \mathcal{T}^c$ fulfils the properties of Eq. (4), then $-\tau \in \mathcal{T}^c$ fulfils the properties of Eq. (4) as well.

The statement $\omega_1 \prec \ldots \prec \omega_c$ is equivalent to the statement $\omega_c \prec \ldots \prec \omega_1$. Each ordinal arrangement of the classes has two edges. Each of the two edges can be seen as the *starting point* of the corresponding order. Therefore, the *uniqueness* of an ordinal class structure is provided by *two* class order arrangements.

Applying τ to Ω leads to the corresponding ordered label set $\tau(\Omega) := \{\omega_{\tau(1)}, \ldots, \omega_{\tau(c)}\}$. Moreover, we define the classes $\omega_{\tau(1)}$ and $\omega_{\tau(c)}$, in an (SVM-)ordered label set $\tau(\Omega)$, as *edge classes*, or simply *edges*, if the context is clear.

C. INTERPRETATION

Let us assume, that according to Definition 1, the labelled data set $X \subset \mathbb{R}^d$ constitutes an SVM-ordinal c-class classification task, with respect to the initial order of the label set Ω. Then, the structure of the corresponding PAM, $\mathcal{A}_\Omega = \mathcal{A}_\Omega^{(id)}$, fulfils the properties of Equation (4), and can be hence depicted as in Equation (3).

The elements of the first row are monotonously increasing, whereas the elements of the last row are monotonously decreasing. For each other row vector of the matrix $\mathcal{A}_\Omega$, the following properties hold. The elements are monotonously decreasing, from the first element to the diagonal element. The elements are monotonously increasing, from the diagonal element to the last element. Since each PAM $\mathcal{A}_\Omega^{(\tau)}$ is symmetric, for any $\tau \in \mathcal{T}^c$, the same properties hold for the corresponding column vectors. Therefore, Equation (4) is equivalent to

$$\begin{array}{rl} a_{j,i}^{(\tau)} \geq a_{k,i}^{(\tau)} & \forall j < k \leq i, \\ \wedge \quad a_{j,i}^{(\tau)} \leq a_{k,i}^{(\tau)} & \forall i \leq j < k. \end{array}$$

The matrix element $a_{i,j}$ can be interpreted as a kind of an *answer* to the question, "How good can the class ω_i be separated from the class ω_j?" In general, it is to assume that, the edge classes can be best separated from each other, compared to any other possible combination of class pairs. Let us assume that the classes ω_1 and ω_c are identified as the edges in an (SVM-)ordinal classification task. Then, it is to expect that it holds, $a_{1,c} \geq a_{i,j} \; \forall i, j = 1, \ldots, c$. That means that the maximum value of the matrix $\mathcal{A}_\Omega$ is identified as the element that is located in the upper right corner, and hence, due to the symmetry of the matrix, in the lower left corner.

Note that equivalently, it is possible to use *pairwise error matrices* (PEMs). In the case of using PEMs, one simply has to define the diagonal elements as ∞, and replace the desired relations "$\geq$" by "$\leq$", and vice versa.

D. DISCUSSION - WHY USING SVMs?

Our proposed working definition for SVM-ordinal classification tasks is based on the resubstitution accuracies, and therefore decision boundaries, which are provided by the corresponding binary SVMs. It is a well-known fact that resubstitution accuracies tend to be too optimistic. Therefore, we would like to clarify the following question. Why did we choose to define SVM-ordinal classification tasks based on resubstitution accuracies, in the first place?

First, the detection of (SVM-)ordinal structures is not part of the training phase of a classification task. It is part of the data analysis. Note that in the current setting, we do not apply the SVMs as classification models, in the classical sense. We use the functionality of the SVMs to check for specific properties of the given data. In a real-world application, there is no knowledge about the labels of the test data. Only the training data is used to analyse possible ordinal class structures. Therefore, in the current setting, we do not need to build an architecture with a high generalisation ability. In contrast, we must explore the whole training data to be able to make reliable statements about its properties in regard to (SVM-)ordinal structures. For the detection of (SVM-)ordinal structures, it is essential to get an *accurate localisation* of the given classes, which is described by the hyperplanes that are provided by the SVMs. Once an (SVM-)ordinal structure is found, one can implement strong ordinal classification models, during the actual training process.

Second, while classification models, such as *unpruned* decision trees [5], can *easily* overfit to the training data, and hence achieve a resubstitution accuracy of 100%, we chose SVMs with linear kernels. Note that each SVM maximises the margin, during the training process (see Sec. II-A). Therefore, if the given binary subtasks are not linearly separable, then the corresponding resubstitution accuracies will differ from 100%. In fact, depending on the *location* in the feature space, of the corresponding binary subtasks, each SVM can lead to any accuracy value, significantly different from 100%.

Third, the fact that the PAM values might be overoptimistic, is not an issue for the following reason. The absolute values of the PAMs are not important. SVM-ordinal structures are identified based on the *relations* $(\leq, \geq)$ between the PAMs elements.

Moreover, SVMs are deterministic models, i.e. identical data sets lead to equal hyperplanes. This observation is an important feature of our proposed working definition. It ensures the reproducibility of our definition. Therefore, it does not make sense to provide a definition based on some kind of hold-out or cross-validation evaluations.

Note that it is possible to generalise our proposed definition by changing the classification model, i.e. by denoting our provided definition as *CM-ordinal*, whereby CM denotes the chosen classification model. However, based on the discussion above, we propose to focus on linear SVMs, since they are non-overfitting and deterministic models, in general.

E. DISCUSSION - THE MEANING OF CLASS LABELS

In the traditional sense, a classification task is defined as ordinal, based on the meaning of class labels, in general. Note that in our proposed definition, we discard/ignore the meaning of class labels completely. This fact might sound questionable and unconventional, at first sight. However, Lattke *et al.* [22] and Lausser *et al.* [23] showed that an expected class structure, which is based on the meaning of classes, is not always reflected in the chosen feature space. In such classification tasks, OC-based classification models (CMs) are not able to exploit the given feature space, with respect to the assumed class order, in general. On the other hand, OC-based CMs might improve the classification performance by exploiting an SVM-ordinal class structure in the given feature space, even for data sets whose class labels do not seem to present an intuitive class order.

As an example assume that we have a set of three classes, i.e. $\Omega = \{$*Human, Hamster, Galápagos Tortoise*$\}$. If we chose the feature *hight*, we would order the set of class labels according to *Hamster* $\prec$ *Galápagos Tortoise* $\prec$ *Human*. Based on the feature *life expectancy*, we would order the set of class labels according to *Hamster* $\prec$ *Human* $\prec$ *Galápagos Tortoise*.

F. THEOREM FOR 3-CLASS CLASSIFICATION TASKS

From our proposed working definition for SVM-ordinal classification tasks, we can draw the following conclusion, for 3-class classification tasks.

Theorem 1 (3-Class Classification Tasks): Let $X \subset \mathbb{R}^d$, $d \in \mathbb{N}$, be a d-dimensional labelled data set, which constitutes a 3-class classification task, i.e. $c = 3$.

Moreover, let the PAM for the set Ω be defined as follows,

$$\mathcal{A}_\Omega^{(id)} = \begin{pmatrix} 0 & e & f \\ e & 0 & g \\ f & g & 0 \end{pmatrix}, \quad e, f, g \in (0, 1].$$

If e, f, g are pairwise distinct, i.e. $e \neq f, e \neq g, f \neq g$, then X constitutes an SVM-ordinal classification task.

The proof of Theorem 1 is provided in the Appendix. By Theorem 1, 3-class classification tasks constitute SVM-ordinal classification tasks, in general. Therefore, for many 3-class classification tasks, one can apply any of the existing tools, from the field of ordinal classification.

As we discussed above, it makes sense to define (SVM-)ordinal class structures based solely on the provided feature space, by disregarding the meaning of the current class labels. Let us think of a two-dimensional 3-class data set. Based on the location of the data points in a two-dimensional feature space, in most cases, it is easy to *identify* two of the class groups as the edges. Moreover, as we discussed above, even a traditionally non-ordinal class label set, such as $\Omega = \{$*Human, Hamster, Galápagos Tortoise*$\}$, can be ordered in different ways, in combination with the chosen feature space.

In the following section, we introduce an algorithm, which provides an easy and effective way to detect SVM-ordinal structures, based on our proposed definition.

IV. DETECTION OF SVM-ORDINAL STRUCTURES

In the current section, we first provide a simple method for the detection of SVM-ordinal structures. Subsequently, we illustrate our proposed detection algorithm based on a simple example.

```
Input: Data set $X \subset \mathbb{R}^d$,
       Label set $\Omega = \{\omega_1, \ldots, \omega_c\}$
Initialisation: PAM $\mathcal{A}_\Omega := \mathbf{0}^{c \times c}$,
       Permutation set $T := \{\}$
  1) FOR $i = 1, \ldots, c-1$ & $j = i+1, \ldots, c$
       • Compute $acc_{i,j}$ according to Eq. (1)
       • Define $a_{i,j}$ according to Eq. (2), i.e.
         $a_{i,j} = acc_{i,j}$ & $a_{j,i} = acc_{i,j}$
  2) FOR $k = 1, \ldots, c$
       • $a_k := (a_{k,1}^{(id)}, \ldots, a_{k,c}^{(id)})$, with $a_{k,k}^{(id)} = 0$
       • Define $\tau_k$, such that $a_k$ is sorted, i.e.
         $a_{k,1}^{(\tau_k)} \leq \ldots \leq a_{k,c}^{(\tau_k)}$
       • IF $\mathcal{A}_\Omega^{(\tau_k)}$ fulfils Eq. (4)
           $T = T \cup \{\tau_k\}$
       • IF $a_k = \left(a_{k,1}^{(id)}, \ldots, a_{k,c}^{(id)}\right)$ includes ties
           Check all $\tilde{\tau}_k$, for which $a_k$ is sorted.
       • IF $|T| = 3$
           BREAK
Output: Set of permutations $T$
```

FIGURE 1. Detection of SVM-ordinal structures. If the given task (X, Ω) constitutes an SVM-ordinal classification task, then the output includes exactly two permutations, which represent the ordinal structure of the current task.

A. DETECTION - ALGORITHM

The pseudo code of our proposed algorithm is depicted in Figure 1. In the first step, i.e. in the first FOR-loop, we simply compute the PAM $\mathcal{A}_\Omega$, according to Eq. (2), with respect to an initial/arbitrary order of the class labels $\Omega = \{\omega_1, \ldots, \omega_c\}$. The second step consists of solely one single FOR-loop, which iterates from 1 to the number of classes c. In each iteration, $k \in \{1, \ldots, c\}$, we take the k-th row of the initial PAM $\mathcal{A}_\Omega^{(id)}$, and define a permutation τ_k which sorts the corresponding vector, denoted by a_k, in ascending order. Subsequently, we apply permutation τ_k to the initial PAM $\mathcal{A}_\Omega^{(id)}$, and check whether the corresponding matrix

$\mathcal{A}_\Omega^{(\tau_k)}$ fulfils the properties of Eq. (4). Note that each sorted vector $a_k^{(\tau_k)}$ represents the first row of the *symbolic matrix* defined in Eq. (3). Therefore, in each step, $k \in \{1, \dots, c\}$, by checking Eq. (4), we analyse whether the class ω_k can be identified as one of the edges. Additionally, in the occurrence of ties, we have to check all corresponding permutations. Moreover, we can stop the detection algorithm if more than two SVM-ordinal permutations were found, since this would indicate that no clear order can be identified.

Therefore, the detection algorithm for SVM-ordinal structures, which is depicted in Figure 1, provides exactly two permutations, for SVM-ordinal classification tasks. This is due to the following fact, which we already discussed in the previous section. The statement $\omega_1 \prec \dots \prec \omega_c$ is equivalent to the statement $\omega_c \prec \dots \prec \omega_1$. Each ordinal arrangement of the classes has two edges. Each of the two edges can be seen as the starting or end point of the corresponding order.

The second FOR-loop of our proposed algorithm constitutes the main difference to the detection methods from [22] and [23]. In both of the related works, one has to consider all possible permutations of the class labels, in general. In contrast, by applying our proposed definition, for each of the classes one has to analyse only one permutation, in general (i.e. if no ties occur). Moreover, in [22] and [23], each permutation is associated with the (re-)evaluation of the corresponding cascaded classification model. By contrast, in our detection algorithm, we evaluate each of the $c(c-1)/2$ binary SVM models exactly one time (in the first FOR-loop).

B. DETECTION - EXAMPLE

We conclude the current section with an illustration of our provided detection algorithm for SVM-ordinal structures. Let us assume that the data set $X \subset \mathbb{R}^d$ constitutes a 4-class classification task, with the corresponding set of labels, $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$. Moreover, let us assume that the computation of the PAM leads to the following matrix, with respect to the initial order of the label set Ω,

$$\mathcal{A}_\Omega^{(id)} \times 100 = \begin{array}{c} \begin{matrix} \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{matrix} \\ \begin{pmatrix} 0 & 83 & 91 & 77 \\ 83 & 0 & 95 & 94 \\ 91 & 95 & 0 & 75 \\ 77 & 94 & 75 & 0 \end{pmatrix} \end{array}.$$

$\underline{k=1}$ $\quad a_1 := (0, 83, 91, 77)$

It holds, $0 \leq 77 \leq 83 \leq 91$, i.e. $a_{1,\mathbf{1}} \leq a_{1,\mathbf{4}} \leq a_{1,\mathbf{2}} \leq a_{1,\mathbf{3}}$. Therefore, we define permutation τ_1 by $\tau_1 : \{1, 2, 3, 4\} \mapsto \{\mathbf{1}, \mathbf{4}, \mathbf{2}, \mathbf{3}\}$,

$$\rightsquigarrow \mathcal{A}_\Omega^{(\tau_1)} \times 100 = \begin{array}{c} \begin{matrix} \boldsymbol{\omega_1} & \boldsymbol{\omega_4} & \boldsymbol{\omega_2} & \boldsymbol{\omega_3} \end{matrix} \\ \begin{pmatrix} 0 & 77 & 83 & 91 \\ 77 & 0 & 94 & 75 \\ 83 & 94 & 0 & 95 \\ 91 & 75 & 95 & 0 \end{pmatrix} \end{array}. \quad ↯$$

Matrix $\mathcal{A}_\Omega^{(\tau_1)}$ does not fulfil the properties of Equation (4). For example, we can observe that, the last row vector of $\mathcal{A}_\Omega^{(\tau_1)}$ is not monotonously decreasing.

$\underline{k=2}$ $\quad a_2 := (83, 0, 95, 94)$

It holds, $0 \leq 83 \leq 94 \leq 95$, i.e. $a_{2,\mathbf{2}} \leq a_{2,\mathbf{1}} \leq a_{2,\mathbf{4}} \leq a_{2,\mathbf{3}}$. Therefore, we define permutation τ_2 by $\tau_2 : \{1, 2, 3, 4\} \mapsto \{\mathbf{2}, \mathbf{1}, \mathbf{4}, \mathbf{3}\}$,

$$\rightsquigarrow \mathcal{A}_\Omega^{(\tau_2)} \times 100 = \begin{array}{c} \begin{matrix} \boldsymbol{\omega_2} & \boldsymbol{\omega_1} & \boldsymbol{\omega_4} & \boldsymbol{\omega_3} \end{matrix} \\ \begin{pmatrix} 0 & 83 & 94 & 95 \\ 83 & 0 & 77 & 91 \\ 94 & 77 & 0 & 75 \\ 95 & 91 & 75 & 0 \end{pmatrix} \end{array}. \quad \checkmark$$

The pairwise accuracy matrix $\mathcal{A}_\Omega^{(\tau_2)}$ fulfils the properties of Equation (4). Therefore, the output set of permutations is extended by τ_2, i.e. $T = \emptyset \cup \tau_2 = \{\tau_2\}$.

$\underline{k=3}$ $\quad a_3 := (91, 95, 0, 75)$

It holds, $0 \leq 75 \leq 91 \leq 95$, i.e. $a_{3,\mathbf{3}} \leq a_{3,\mathbf{4}} \leq a_{3,\mathbf{1}} \leq a_{3,\mathbf{2}}$. Therefore, we define permutation τ_3 by $\tau_3 : \{1, 2, 3, 4\} \mapsto \{\mathbf{3}, \mathbf{4}, \mathbf{1}, \mathbf{2}\}$,

$$\rightsquigarrow \mathcal{A}_\Omega^{(\tau_3)} \times 100 = \begin{array}{c} \begin{matrix} \boldsymbol{\omega_3} & \boldsymbol{\omega_4} & \boldsymbol{\omega_1} & \boldsymbol{\omega_2} \end{matrix} \\ \begin{pmatrix} 0 & 75 & 91 & 95 \\ 75 & 0 & 77 & 94 \\ 91 & 77 & 0 & 83 \\ 95 & 94 & 83 & 0 \end{pmatrix} \end{array}. \quad \checkmark$$

The pairwise accuracy matrix $\mathcal{A}_\Omega^{(\tau_3)}$ fulfils the properties of Equation (4). Therefore, the output set of permutations is extended by τ_3, i.e. $T = \{\tau_2\} \cup \tau_3 = \{\tau_2, \tau_3\}$.

$\underline{k=4}$ $\quad a_4 := (77, 94, 75, 0)$

It holds, $0 \leq 75 \leq 77 \leq 94$, i.e. $a_{4,4} \leq a_{4,\mathbf{3}} \leq a_{4,\mathbf{1}} \leq a_{4,\mathbf{2}}$. Therefore, we define permutation τ_4 by $\tau_4 : \{1, 2, 3, 4\} \mapsto \{\mathbf{4}, \mathbf{3}, \mathbf{1}, \mathbf{2}\}$,

$$\rightsquigarrow \mathcal{A}_\Omega^{(\tau_4)} \times 100 = \begin{array}{c} \begin{matrix} \boldsymbol{\omega_4} & \boldsymbol{\omega_3} & \boldsymbol{\omega_1} & \boldsymbol{\omega_2} \end{matrix} \\ \begin{pmatrix} 0 & 75 & 77 & 94 \\ 75 & 0 & 91 & 95 \\ 77 & 91 & 0 & 83 \\ 94 & 95 & 83 & 0 \end{pmatrix} \end{array}. \quad ↯$$

Matrix $\mathcal{A}_\Omega^{(\tau_4)}$ does not fulfil the properties of Equation (4). For example, we can observe that, the last row vector of $\mathcal{A}_\Omega^{(\tau_1)}$ is not monotonously decreasing.

Since it holds $|T| = 2$, it follows that data set X constitutes an SVM-ordinal classification task. Moreover, note that permutation τ_3 represents the reversed order of permutation τ_2, i.e. $\pm\tau_3 = \mp\tau_2$. Therefore, in the current example, the edge classes are clearly identified as ω_2 and ω_3.

V. EVALUATION OF ORDINAL DATA SETS

In the current section, we evaluate a set of benchmark data sets from the field of ordinal classification. All of the data sets are publicly available. We include the data sets that have been also analysed in [19, Table 4]. Note that by Theorem 1, 3-class

TABLE 1. Properties of the traditionally ordinal data sets. #F: Number of total features (number of categorical features). #C: Number of classes. #S: Number of Samples. ω_i: Number of samples in class i.

Data Set	#F	#C	#S	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9
CMC	9 (7)	3	1473	629	511	333	–	–	–	–	–	–
Grub Damage	8 (6)	4	155	49	41	46	19	–	–	–	–	–
LEV-4	4 (0)	4	1000	93	280	403	224	–	–	–	–	–
SWD	10 (0)	4	1000	32	352	399	217	–	–	–	–	–
Car Evaluation	6 (6)	4	1728	1210	384	69	65	–	–	–	–	–
Nursery	8 (8)	4	12958	4320	328	4266	4044	–	–	–	–	–
ESL-5	4 (0)	5	488	52	100	116	135	85	–	–	–	–
LEV	4 (0)	5	1000	93	280	403	197	27	–	–	–	–
ESL	4 (0)	9	488	2	12	38	100	116	135	62	19	4
ERA	4 (0)	9	1000	92	142	181	172	158	118	88	31	18

classification tasks constitute SVM-ordinal classification tasks, in general. Therefore, we focus mainly on the tasks with at least four classes.

A. DATA SETS - AVAILABILITY

From the UCI machine learning repository [13], we included the Contraceptive Method Choice data set, the Car Evaluation data set, as well as the Nursery data set. The remaining data sets are all available on the Weka website.[1] The Grub Damage data set is included in the file denoted by *agridatasets.jar*. The data sets ERA, ESL, LEV and SWD, which were provided by A.B. David, are all included in the file named *datasets-arie_ben_david.tar.gz*.

B. DATA SETS - DESCRIPTION

Table 1 summarises the properties of all of the following ten data sets including the corresponding class distributions.

1) CONTRACEPTIVE METHOD CHOICE (CMC)

This data set is part of the National Indonesia Contraceptive Prevalence Survey from 1987. The task is to predict a married woman's contraceptive method, labelled as *no use* (ω_1), *short-term method* (ω_2), and *long-term method* (ω_3). The features describe the women's demographic and socio-economic characteristics, such as the *age*, *education level*, and *media exposure*.

2) CAR EVALUATION (CARS)

This data set constitutes a car evaluation task, which is based on features such as *price*, *comfort*, and *safety*. The samples are labelled according to the following classes, *unacceptable* (ω_1), *acceptable* (ω_2), *good* (ω_3) and *very good* (ω_4).

3) NURSERY

This data set contains nursery applications, which were evaluated by features such as *family structure and financial standing*, and *social and health picture of the family*. The class labels of the Nursery data set correspond to the final evaluation of the current application, representing one of the following recommendations for the acceptance of the applicant, *not recommended* (ω_1), *recommended*, *very much recommended* (ω_2), *priority acceptance* (ω_3) and *special priority acceptance* (ω_4). Since the class corresponding to *recommended* includes only two samples, we discarded this class completely, leading to a four-class classification task.

4) GRUB DAMAGE

This data set consists of agriculture-specific features for the task of pasture damage estimation, which is related to the number of grass grubs. The classes of the Grub Damage data set are denoted by *low* (ω_1), *average* (ω_2), *high* (ω_3) and *very high* (ω_4).

5) SOCIAL WORKERS DECISIONS (SWD)

This data set consists of the assessments of qualified social workers evaluating the risk facing children, if they stayed with their families. The classes, i.e. risk levels, are simply denoted by the numbers 1, 2, 3, 4.

6) LECTURERS EVALUATION (LEV)

The LEV data set consists of the evaluation of lecturers of MBA courses. The class labels, i.e. the final scores, are simply denoted by 1, 2, 3, 4, 5. Initially, the LEV data set constitutes a 5-class classification task. However, due to the skewness of the class distribution, some authors, e.g. in [22], summarise the two classes, corresponding to the scores 4 and 5, to one resulting class. We will evaluate both variants of the LEV data set, denoting the 4-class LEV data set by LEV-4.

7) EMPLOYEE SELECTION (ESL)

The ESL data set contains the profile applicants for certain industrial jobs. The classes represent a degree of fitness of the applicant to the corresponding type of job. The labels are simply denoted by $1, \ldots, 9$. Initially, the ESL data set constitutes a 9-class classification task. For the same reason as we mentioned above for the LEV data set, some authors summarise this data set to a 5-class classification task. Thereby, the classes specific to the first three scores are fused to one single class, and the classes specific to the last three scores are fused to one single class. We denote the resulting data set by ESL-5.

8) EMPLOYEE REJECTION\ACCEPTANCE (ERA)

The ERA data set is similar to the ESL data set. It also covers an application evaluation task. However, the class labels were

[1] https://waikato.github.io/weka-wiki/datasets/

not defined by expert recruiters, but were acquired during an academic MBA course. The labels are again simply denoted by $1, \ldots, 9$, representing the scores for the tendency to reject or accept the applicant.

C. RESULTS

We implemented linear SVMs in combination with the Sequential Minimal Optimization (SMO) solver [16], [20]. For each of the ten data sets from Table 1, we applied our proposed detection algorithm, which is presented in Figure 1. Seven out of the ten data sets are identified as SVM-ordinal by Definition 1, i.e. only the corresponding matrices $\mathcal{A}_\Omega^{(id)}$, $\mathcal{A}_\Omega^{(-id)}$ fulfil the properties of Equation (4), with respect to the natural order of the classes. The following data sets were not identified as SVM-ordinal by our proposed definition, Nursery, ESL, and ERA.

Note that in Section IV, we presented our SVM-ordinal structure detection algorithm. Moreover, in the example, in Sec. IV-B, we permuted the initial order of the LEV-4 data set to illustrate our proposed detection algorithm. The PAMs of the aforementioned example, which fulfil the properties of Equation (4), represent the natural order of the LEV-4 data set.

D. DISCUSSION

In our validation experiments, seven out of ten benchmark data sets were identified as SVM-ordinal, by our proposed working definition, with respect to the natural order of the classes. What does that mean in regard to our proposed definition? According to our interpretation, the results confirm the validity of our proposed working definition for SVM-ordinal classification tasks. Although the data sets Nursery, ESL, and ERA were not identified as SVM-ordinal, that does not violate the validity of our proposed definition, for the following reasons. All of the data sets from Table 1 are assumed to be ordinal *by convention*. So far, there exists no fundamental theory, which leads to a proof or contradiction of that assumption. Therefore, the fact that some of the data sets were not identified as SVM-ordinal, does not necessarily indicate a weakness or shortcoming of our proposed working definition. In contrast, this is an accepted outcome of our study.

A natural consequence of our proposed definition is that a data set, whose class labels exhibit a clear natural order (i.e. an OC task in the traditional sense) may not exhibit the considered structure in the provided feature space, and hence are not defined as SVM-ordinal by our proposed definition. This observation is also supported by Lattke *et al.* in [23], where the authors conclude that the ordinal characteristics might not be reflected in the chosen feature space. As discussed in Section III, defining a multi-class task as ordinal, based solely on the meaning of its class labels, is not useful if the ordered class structure is not reflected in the feature space. Since each classification model is trained in combination with the chosen feature space, an OC-based classifier can only benefit from a present feature space ordinal class structure.

VI. EVALUATION OF NON-ORDINAL DATA SETS

In the current section, we evaluate five additional, traditionally non-ordinal, data sets from the UCI machine learning repository.

A. DATA SETS DESCRIPTION

The properties of the data sets, which are briefly described below, are summarised in Table 2.

TABLE 2. Properties of the traditionally non-ordinal data sets. #F: Number of features. #C: Number of classes. #S: Number of Samples.

Data Set	#F	#C	#S	Distribution
Seeds	7	3	210	70 per class
Forests	27	4	523	$83 - 86 - 159 - 195$
Vehicles	18	4	846	$199 - 212 - 217 - 218$
Segment	19	7	2310	330 per class
Mfeat	649	10	2000	200 per class

1) SEEDS

This data set consists of images of the internal kernels specific to three types (classes) of wheat, i.e. *Kama*, *Rosa*, and *Canadian*. The feature space consists of seven numerical parameters, including the *length* and the *width* of the kernel, amongst others.

2) FOREST TYPE MAPPING (FORESTS)

The Forests data set consists of four different forest types, which are described by their spectral characteristics at visible-to-near infrared wavelengths. Thus, all of the 27 provided features are numerical. The classes are denoted as *Sugi*, *Hinoki*, *Mixed Deciduous*, and *Non-Forest*, including 195, 86, 159 and 83 samples, respectively.

3) STATLOG VEHICLE SILHOUETTES (VEHICLES)

This data set consists of continuous features, extracted from two-dimensional silhouettes and from different angles, specific to four different vehicles (classes). The experimenters included a double decker bus, a Chevrolet van, a Saab 9000, as well as an Opel Manta 400. The resulting classes are simply denoted by *Opel*, *Saab*, *Bus*, and *Van*.

4) STATLOG IMAGE SEGMENTATION (SEGMENT)

The Segment data set consists of hand-segmented outdoor images from seven different categories (classes). The provided classes are denoted by *Brickface*, *Sky*, *Foliage*, *Cement*, *Window*, *Path*, and *Grass*.

5) MULTIPLE FEATURES (MFEAT)

The Mfeat data set consists of handwritten digits. The features were extracted specific to six different approaches, namely Fourier coefficients of the character shapes, profile correlations, Karhunen-Loeve coefficients, pixel averages in

2×3 windows, Zernike moments, as well as morphological features. This data set constitutes a naturally occurring 10-class classification task (digits $0, \dots, 9$).

B. RESULTS & DISCUSSION

In the current section, we included another data set that constitutes a 3-class classification task, i.e. Seeds. The Seeds data set constitutes an SVM-ordinal classification by our proposed definition, according to the class label order *Rosa* $\prec$ *Kama* $\prec$ *Canadian*. This outcome is a further example for Theorem 1. From Table 2, also the data set Forests was identified as SVM-ordinal, according to the class label order *Hinoki* $\prec$ *Sugi* $\prec$ *Mixed Deciduous* $\prec$ *Non-Forest*.

The results from Section V, as well as from the current section, support our initial thoughts which we discussed with respect to our proposed definition (see Sec. III). There exist traditionally defined ordinal classification tasks (based on the meaning of class labels) that constitute a natural order of the class structure which is not reflected in the feature space. However, by contrast, there also exist multi-class data sets, whose class structures can be ordered, independently from the meaning of the current class labels. As we discussed above, this also implies that the detection of SVM-ordinal class structures, which is based on our proposed definition, is also independent from any kind of task-related expert knowledge.

VII. DETECTION COMPLEXITY & LIMITATIONS

In the current section, we will first discuss the operational cost for the detection of SVM-ordinal structures. Subsequently, we will provide a brief discussion on the limitations of our proposed working definition.

A. OPERATIONAL COST

In the current study, all presented experiments were conducted using an Intel Core i7-6700K @4GHz with Windows7, 64 bit, in combination with the Matlab[2] software. For each data set, we repeated the evaluation of the provided detection algorithm of SVM-ordinal structures (see Fig. 1) ten times. Table 3 states the averaged time values in seconds. The duration for the search of SVM-ordinal class structures depends on the number of samples, classes and features. It is safe to assume that the duration also depends on the *complexity* of the current classification task, i.e. on the resulting number of support vectors of the corresponding $c(c-1)/2$ binary SVM models. However, from Table 3, we can conclude that, in general, the time needed for the identification of SVM-ordinal classification tasks, which we defined in Definition 1, is negligible.

As discussed in the current section, as well as in Section IV, the complexity associated to the design and evaluation of the classification models is reduced by our proposed detection algorithm. In contrast to $\mathcal{O}(c!)$ (exhaustive search), our method leads to a complexity of $\mathcal{O}(c^2)$. The complexity of the additional sorting of each row of the initial PAM can be estimated as $\mathcal{O}(c^2 \log(c))$, since it is the average case for many existing sorting algorithms. Note that, the classification complexity $\mathcal{O}(c^2)$ also depends on the number of features and the number of samples. In contrast, the sorting complexity $\mathcal{O}(c^2 \log(c))$, specific to the rows of the initial PAM, can be neglected, since (SVM-)ordinal classification tasks consists of a *small* number of classes, in general not more than ten classes.

TABLE 3. Computational Time in Seconds for the Detection of SVM-ordinal Structures. #F: Number of features. #C: Number of classes. #S: Number of Samples. For each data set, we repeated the evaluation of the detection algorithm 10 times, and stated the resulting averaged value.

Data Set	#F	#C	#S	Computational Time
Seeds	7	3	210	0.02
CMC	9	3	1473	2.12
Grub Damage	8	4	155	0.06
Forests	27	4	523	2.27
Vehicles	18	4	846	9.06
LEV-4	4	4	1000	0.07
SWD	10	4	1000	0.11
Car Evaluation	6	4	1728	0.13
Nursery	8	4	12958	1.68
ESL-5	4	5	488	0.05
LEV	4	5	1000	0.09
Segment	19	7	2310	13.95
ESL	4	9	488	0.19
ERA	4	9	1000	0.60
Mfeat	649	10	2000	19.55

Moreover, note that, in accordance to our discussion in Section III, each of the ten evaluations of the detection algorithm led to exactly the same PAMs, specific to each of the data sets from Table 3 (due to the fact that SVMs are deterministic models).

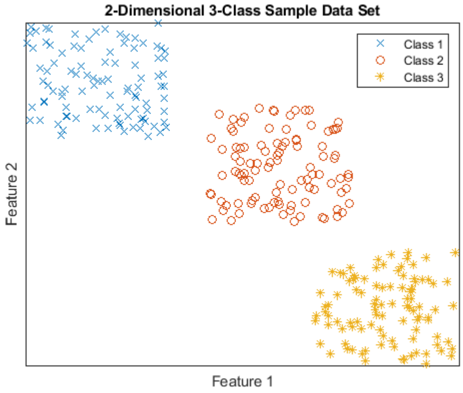

FIGURE 2. Artificially Constructed Ordinal Toy Data Set.

B. LIMITATIONS

One shortcoming of our proposed definition is that it is not possible to detect SVM-ordinal structures of data sets whose classes are linearly separable. Figure 2 depicts a two-dimensional 3-class data set whose classes are linearly separable. According to Figure 2, the (SVM-)ordinal structure of the provided data set is clear, i.e. *Class 1* $\prec$ *Class 2* $\prec$ *Class 3*. However, for all $\tau \in \mathcal{T}^3$, the computation

[2] https://www.mathworks.com/products/matlab.html

of the corresponding PAMs $\mathcal{A}_{\Omega}^{(\tau)}$ leads to

$$\mathcal{A}_{\Omega}^{(\tau)} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Therefore, each possible permutation of the classes represents an ordinal class structure. Hence, the data set is not SVM-ordinal by our proposed definition, since no clear class structure can be identified, i.e. the uniqueness property of Definition 1 is violated. However, the currently discussed shortcoming does not constitute a serious issue, for the following reasons. First, classification tasks which include linearly separable classes are *well-posed*, i.e. in general, such tasks are *easy* to solve by any kind of classification models. Hence there is no need to implement OC-based classifiers. Second, it might be more useful to evaluate such kind of data sets in combination with different cluster analysis techniques, instead of different classification approaches.

A second shortcoming of our proposed definition is that it is not applicable for classification tasks, for which the accuracy/error rate is not an appropriate performance measure. However, in general, it should be possible to adapt our proposed definition to the currently desired application-dependent measure.

VIII. CONCLUSION

In the current study, we proposed a *working definition* for ordinal classification (OC) tasks. We use the term ''working definition'', because our proposed definition is based on the provided hyperplanes of standard SVM models. As a consequence, in contrast to the traditional interpretation of OC tasks, in our proposed definition, we completely discard the meaning of class labels and focus solely on the current feature space. Therefore, we introduced the term *SVM-ordinal* class structures, to differentiate between traditionally ordinal OC tasks and SVM-ordinal OC tasks, with respect to our proposed definition.

The advantages of our proposed definition can be summarised as follows. First, our proposed definition can be interpreted easily, since it originates from the following intuitive idea. It is easier to differentiate between two *distant* classes than between two *near* classes. Therefore, if a data set constitutes an SVM-ordinal classification task, the following property must hold. If we select and fix one class, and try to separate it from a second class, then the corresponding binary classification task becomes easier to solve, each time the second class is chosen nearer to one of the edges. The second advantage of our proposed definition is that the definition is easy to apply since it is based on the decision boundaries of standard SVMs, which are part of the most commonly used software tools, such as Matlab, GNU Octave, Python, Weka, Ruby and R, amongst others.

Based on our proposed definition, we introduced a simple method for the detection of SVM-ordinal structures. Therefore, SVM-ordinal structures can be found without any additional expert knowledge, and even without any additional meta information about the corresponding classification task. Moreover, we showed that according to our proposed definition, 3-class classification tasks can be identified as SVM-ordinal classification tasks, in general. Therefore, for many 3-class classification tasks, one can apply one of the existing ordinal classification specific techniques.

In the current study, we included ten multi-class data sets, which are used as benchmark data sets, in the field of ordinal classification, as well as five data sets, which are non-ordinal in the traditional sense. We analysed all of the classification tasks, in combination with our proposed definition. The evaluations of the included data sets were discussed in detail and support the validity of our proposed definition.

Finally, we believe that we provided a simple and interpretable definition, which constitutes a supporting tool that can be useful for researches, from the field of ordinal classification, or multi-class classification in general.

APPENDIX - PROOF OF THEOREM 1

Let $X \subset \mathbb{R}^d, d \in \mathbb{N}$, be a d-dimensional labelled data set, with the corresponding set of class labels $\Omega = \{\omega_1, \omega_2, \omega_3\}$. Let the PAM for the set Ω be defined as follows,

$$\mathcal{A}_{\Omega}^{(id)} = \left(\begin{array}{ccc} \omega_1 & \omega_2 & \omega_3 \\ 0 & e & f \\ e & 0 & g \\ f & g & 0 \end{array}\right), \text{ with } e, f, g \in (0, 1].$$

Moreover, let e, f, g be pairwise distinct, i.e. $e \neq f, e \neq g$, and $f \neq g$.

Existence. **Claim**: There exist permutations $\tau \in \mathcal{T}^3$, such that $\mathcal{A}_{\Omega}^{(\tau)}$ fulfils the properties of Eq. (4).

Proof: If $\mathcal{A}_{\Omega}^{(id)}$ fulfils the properties of Equation (4), then there is nothing to show. Therefore, we now assume that $\mathcal{A}_{\Omega}^{(id)}$ does not fulfil the properties of Equation (4). Then, we obtain the following cases.

Case 1: $e > f$, with either $(f < g)$ or $(f > g)$.

We define permutation τ_1 by $\tau_1 : \{1, 2, 3\} \mapsto \{1, 3, 2\}$. Thus, we obtain the following matrix, with respect to τ_1,

$$\mathcal{A}_{\Omega}^{(\tau_1)} = \left(\begin{array}{ccc} \omega_1 & \omega_3 & \omega_2 \\ 0 & f & e \\ f & 0 & g \\ e & g & 0 \end{array}\right).$$

Case 1.1: $e > g$.

Note that we already assumed that it holds, $e > f$. Therefore, from the relation $e > g$, it directly follows that matrix $\mathcal{A}_{\Omega}^{(\tau_1)}$ fulfils the properties of Equation (4).

Case 1.2: $e < g$.

We define permutation τ_2 by $\tau_2 : \{1, 2, 3\} \mapsto \{2, 1, 3\}$. Thus, we obtain the following matrix, with respect to τ_2,

$$\mathcal{A}_{\Omega}^{(\tau_2)} = \left(\begin{array}{ccc} \omega_2 & \omega_1 & \omega_3 \\ 0 & e & g \\ e & 0 & f \\ g & f & 0 \end{array}\right). \quad (5)$$

In total, in this case it holds, $f < e < g$. Thus, matrix $\mathcal{A}_\Omega^{(\tau_2)}$ fulfils the properties of Equation (4).

Case 2: $f < g$, with either $(e < f)$ or $(e > f)$.

Again, we apply $\tau_2 : \{1, 2, 3\} \mapsto \{2, 1, 3\}$, and obtain the matrix, with respect to τ_2 which is defined in Equation (5).

Case 2.1: $e < g$.

Note that we already assumed that it holds, $f < g$. Therefore, from the relation $e < g$, it directly follows that matrix $\mathcal{A}^{(\tau_2)}$ fulfils the properties of Equation (4).

Case 2.2: $e > g$.

We define permutation τ_3 by $\tau_3 : \{1, 2, 3\} \mapsto \{2, 3, 1\}$. Thus, we obtain the following matrix, with respect to τ_3,

$$\mathcal{A}_\Omega^{(\tau_3)} = \begin{array}{c} \begin{matrix} \omega_2 & \omega_3 & \omega_1 \end{matrix} \\ \begin{pmatrix} 0 & f & e \\ f & 0 & g \\ e & g & 0 \end{pmatrix} \end{array},$$

In total, in this case it holds $f < g < e$. Thus, matrix $\mathcal{A}_\Omega^{(\tau_3)}$ fulfils the properties of Equation (4).

Note that the Cases 1 and 2, including the corresponding sub-cases, also cover the case $(e > f) \wedge (f < g)$. Therefore, we covered all possible cases for the initial PAM $\mathcal{A}_\Omega^{(id)}$. Hence, there always exist a permutation $\tau \in \mathcal{T}^3$, such that $\tau(\Omega)$ represents an ordinal class structure.

Uniqueness. **Claim**: There exist exactly two permutations $\tau \in \mathcal{T}^3$, such that $\mathcal{A}_\Omega^{(\tau)}$ fulfils the properties of Equation (4).

Proof: Without loss of generality, we assume that the PAM

$$\mathcal{A}_\Omega^{(id)} = \begin{array}{c} \begin{matrix} \omega_1 & \omega_2 & \omega_3 \end{matrix} \\ \begin{pmatrix} 0 & e & f \\ e & 0 & g \\ f & g & 0 \end{pmatrix} \end{array},$$

fulfils the properties of Equation (4). By definition, it follows that permutation $-id : \{1, 2, 3\} \mapsto \{3, 2, 1\}$ also fulfils the properties of Equation (4). Moreover, since $\mathcal{A}_\Omega^{(id)}$ fulfils the properties of Equation (4), and since e, f, g are pairwise distinct, it must hold

$$e < f, \quad \text{and} f > g. \tag{6}$$

Therefore, we now have to show that the remaining permutations violate the properties of Equation (4).

Permutations $\tau_1 : \{1, 2, 3\} \mapsto \{1, 3, 2\}$, and $-\tau_1$.

We obtain the following matrix, with respect to τ_1,

$$\mathcal{A}_\Omega^{(\tau_1)} = \begin{array}{c} \begin{matrix} \omega_1 & \omega_3 & \omega_2 \end{matrix} \\ \begin{pmatrix} 0 & f & e \\ f & 0 & g \\ e & g & 0 \end{pmatrix} \end{array}. \quad \lightning$$

PAM $\mathcal{A}_\Omega^{(\tau_1)}$ violates the properties of Eq. (4), since it holds $e < f$ by Eq. (6). Hence, also $-\tau_1 : \{1, 2, 3\} \mapsto \{2, 3, 1\}$ violates the properties of Eq. (4), by definition.

Permutations $\tau_2 : \{1, 2, 3\} \mapsto \{2, 1, 3\}$, and $-\tau_2$.

We obtain the following matrix, with respect to τ_2,

$$\mathcal{A}_\Omega^{(\tau_1)} = \begin{array}{c} \begin{matrix} \omega_2 & \omega_1 & \omega_3 \end{matrix} \\ \begin{pmatrix} 0 & e & g \\ e & 0 & f \\ g & f & 0 \end{pmatrix} \end{array}. \quad \lightning$$

PAM $\mathcal{A}_\Omega^{(\tau_1)}$ violates the properties of Eq. (4), since it holds $f > g$ by Eq. (6). Hence, also $-\tau_2 : \{1, 2, 3\} \mapsto \{3, 1, 2\}$ violates the properties of Eq. (4), by definition. □

In the first part of the proof, we showed that, if the elements of the upper/lower triangular PAM are pairwise distinct, then there exist (at least) two permutations $\tau \in \mathcal{T}^3$, such that the corresponding PAM $\mathcal{A}_\Omega^{(\tau)}$ fulfils the properties of Equation (4). In the second part of the proof, we showed that there exist exactly two permutations $\tau \in \mathcal{T}^3$, such that the corresponding PAM $\mathcal{A}_\Omega^{(\tau)}$ fulfils the properties of Equation (4). Note that $|\mathcal{T}^3| = 6$, i.e. there exist six distinct permutations of the set $\Omega = \{\omega_1, \omega_2, \omega_3\}$. Moreover the six permutations can be grouped into three pairs, which represent one of the possible *unique* class orders each.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. Moreover, They would also like to thank the reviewers for their constructive feedback, which helped to significantly improve the current work.

REFERENCES

[1] S. Abe, "Support vector machines for pattern classification," in *Advances in Pattern Recognition*. London, U.K.: Springer, 2005.

[2] A. Agresti, "*Analysis of Ordinal Categorical Data*, vol. 656. Hoboken, NJ, USA: Wiley, 2010.

[3] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Mach. Learn. Res.*, vol. 1, pp. 113–141, Sep. 2001.

[4] P. Bellmann, P. Thiam, and F. Schwenker, *Multi-Classifier-Systems: Architectures, Algorithms and Applications*. Cham, Switzerland: Springer, 2018, pp. 83–113.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification Regression Trees*, Wadsworth, OH, USA, 1984.

[6] J. S. Cardoso, J. F. Pinto da Costa, and M. J. Cardoso, "Modelling ordinal relations with SVMs: An application to objective aesthetic evaluation of breast cancer conservative treatment," *Neural Netw.*, vol. 18, nos. 5–6, pp. 808–817, Jul. 2005.

[7] J. S. Cardoso and R. Sousa, "Measuring the performance of ordinal classification," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 25, no. 08, pp. 1173–1195, Dec. 2011.

[8] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using ranking-CNN for age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 742–751.

[9] W. Chu and S. S. Keerthi, "New approaches to support vector ordinal regression," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 145–152.

[10] W. Chu and S. S. Keerthi, "Support vector ordinal regression," *Neural Comput.*, vol. 19, no. 3, pp. 792–815, Mar. 2007.

[11] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, Jul. 2014.

[12] T. G. Dietterich and G. Bakiri, "Error-correcting output codes: A general method for improving multiclass inductive learning programs," in *Proc. AAAI*. Cambridge, MA, USA: MIT Press, 1991, pp. 572–577.

[13] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[14] S. Escalera, O. Pujol, and P. Radeva, ''Separability of ternary codes for sparse designs of error-correcting output codes,'' *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 285–297, Feb. 2009.

[15] S. Escalera, O. Pujol, and P. Radeva, ''On the decoding process in ternary error-correcting output codes,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 120–134, Jan. 2010.

[16] R.-E. Fan, P.-H. Chen, and C.-J. Lin, ''Working set selection using second order information for training support vector machines,'' *J. Mach. Learn. Res.*, vol. 6, pp. 1889–1918, Dec. 2005.

[17] E. Frank and M. A. Hall, ''A simple approach to ordinal classification,'' in *Proc. Eur. Conf. Mach. Learn.* (Lecture Notes in Computer Science), vol. 2167. Berlin, Germany: Springer, 2001, pp. 145–156.

[18] P. A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez, ''Ordinal regression methods: Survey and experimental study,'' *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 127–146, Jan. 2016.

[19] J. C. Hühn and E. Hüllermeier, ''Is an ordinal class structure useful in classifier learning?'' *Int. J. Data Mining, Model. Manage.*, vol. 1, no. 1, pp. 45–67, 2008.

[20] M. Kächele, G. Palm, and F. Schwenker, ''SMO lattices for the parallel training of support vector machines,'' in *Proc. ESANN*, 2015, pp. 1–6.

[21] L. I. Kuncheva, *Combining Pattern Classifiers: Methods Algorithms*. Hoboken, NJ, USA: Wiley, 2014, ch. 7, pp. 230–244.

[22] R. Lattke, L. Lausser, C. Müssel, and H. A. Kestler, ''Detecting ordinal class structures,'' in *Multiple Classifier Systems* (Lecture Notes in Computer Science) vol. 9132. Cham, Switzerland: Springer, 2015, pp. 100–111.

[23] L. Lausser, L. M. Schäfer, L.-R. Schirra, R. Szekely, F. Schmid, and H. A. Kestler, ''Assessing phenotype order in molecular data,'' *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, Dec. 2019.

[24] Y. Liu, A. W. K. Kong, and C. K. Goh, ''Deep ordinal regression based on data relationship for small datasets,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2372–2378.

[25] P. McCullagh, ''Regression models for ordinal data,'' *J. Roy. Stat. Soc., B (Methodol.)*, vol. 42, no. 2, pp. 109–127, 1980.

[26] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, ''Ordinal regression with multiple output CNN for age estimation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4920–4928.

[27] D. M. J. Tax and R. P. W. Duin, ''Using two-class classifiers for multiclass classification,'' in *Proc. Object Recognit. Supported Interact. Service Robots*, 2002, pp. 124–127.

[28] P. Thiam, V. Kessler, M. Amirian, P. Bellmann, G. Layher, Y. Zhang, M. Velana, S. Gruss, S. Walter, H. C. Traue, J. Kim, D. Schork, E. Andre, H. Neumann, and F. Schwenker, ''Multi-modal pain intensity recognition based on the SenseEmotion database,'' *IEEE Trans. Affect. Comput.*, early access, Jan. 10, 2019, doi: 10.1109/TAFFC.2019.2892090.

PETER BELLMANN received the degree in mathematics from Ulm University, Ulm, Germany, in 2016, where he is currently pursuing the Ph.D. degree in computer science with the Neural Information Processing Department. He was supported by the scholarship of the Landesgraduiertenförderung Baden-Württemberg, Ulm University. His research interests include ordinal classification, multiple classifier systems, multi-modal fusion architectures, and machine learning techniques for the recognition of affective states in human-centered signals.

FRIEDHELM SCHWENKER (Member, IEEE) received the Diploma and Ph.D. degrees in mathematics and computer science from the University of Osnabrück. He is currently a Privatdozent with the Institute of Neural Information Processing, Ulm University. He has (co-) edited 20 special issues and workshop proceedings published in international journals and publishing companies. He has published more than 200 articles at international conferences and journals. His research interests include artificial neural networks, machine learning, statistical learning theory, data mining, pattern recognition, information fusion, and affective computing. He has serves as the (Co-) Chair for the IAPR TC3 on neural networks and computational intelligence. Since 2016, he has been the Chair of the new IAPR TC9 on pattern recognition in human–computer interaction.