Ordinal Classifiers Can Fail on Repetitive Class Structures

Ludwig Lausser*, Lisa M. Schäfer*, Hans A. Kestler

Abstract Ordinal classifiers are constrained classification algorithms that assume a predefined (total) order of the class labels to be reflected in the feature space of a dataset. This information is used to guide the training of ordinal classifiers and might lead to an improved classification performance. Incorrect assumptions on the order of a dataset can result in diminished detection rates. Ordinal classifiers can, therefore, be used to screen for ordinal class structures within a feature representation. While it was shown that algorithms could in principle reject incorrect class orderings, it is unclear if all remaining candidate orders reflect real ordinal structures in feature space.

In this work we characterize the decision regions induced by ordinal classifiers. We show that they can fulfill different criteria that might be considered as ordinal

Ludwig Lausser* · Hans A. Kestler
Institute of Medical Systems Biology
Ulm University, 89069 Ulm, Germany
✉ ludwig.lausser@uni-ulm.de
✉ hans.kestler@uni-ulm.de

Lisa M. Schäfer*
Institute of Medical Systems Biology and Graduate School 2254 (HEIST)
Ulm University, 89069 Ulm, Germany
✉ lisa-1.schaefer@uni-ulm.de

*equal contribution

Archives of Data Science, Series A
(Online First)
KIT Scientific Publishing
Vol. 4, No. 1, 2018

DOI: 10.5445/KSP/1000085951/25
ISSN 2363-9881

reflections. These criteria are mainly determined by the connectedness and the neighborhood of the decision regions. We evaluate them for ordinal classifier cascades constructed from binary classifiers. We show that depending on the type of base classifier they bear the risk of not rejecting non ordinal, like partial repetitive, structures.

1 Introduction

Classification, as a supervised learning task, is the canonical example for a machine learning technique that bridges the gap between subsymbolic information and semantically meaningful classes (categories, concepts, etc.). By extracting class predicting patterns, these algorithms generate a measurable representation of verbal concepts. These patterns can reveal unknown class properties or causes of events. Nevertheless, their existence cannot be guaranteed. Unsuitable feature representations might lack any information (Lausser et al., 2013; Schirra et al., 2016).

The requirements on a feature representation become even more complex when it is assumed that they reflect semantic relations among the embedded classes (Lausser et al., 2014; Taudien et al., 2016; Lausser et al., 2018). While these relationships are well known for the verbal concept of a class, it is unclear what these relations look like in feature space. They might not be reflected at all. Identifying a feature representation that reflects predefined semantic relations must, therefore, be regarded as a rare event. Nevertheless, it provides much more profound insights into the properties and relationships of the classes. We focus on ordinal semantic relationships between classes, e. g.

$$stage_1 \prec stage_2 \prec stage_3\,, \tag{1}$$

which occur in diverse fields, like medicine (Weinberg, 2013). The ordinal relationship $\prec$ is only known for the verbal concepts $stage_1$, $stage_2$, $stage_3$ (e.g. in medicine defined according to some morphological characteristics); its reflection on the molecular level (feature representation) is not guaranteed. An example might be tumorigenesis, where the definition of stage is often based on histological observations, like the grade of tissue disruption (Hruban et al., 2001). Finding possible orders in the gene expression profile can hereby help to confirm or falsify previous hypotheses.

Classifiers that rely on a predefined class order are constrained classification algorithms that are restricted by their choice of placing decision boundaries. These restrictions can guide the training process of *ordinal classifiers* (Cardoso and Pinto da Costa, 2007). They can also be misleading when an assumed class order deviates from the real class order or embedding in the feature space (Lattke et al., 2015).

Many ordinal classifiers are multi-class architectures of binary base classifiers. Frank and Hall (2001) analyze ordinal classifier cascades, which are adapted decision lists (Rivest, 1987). Hühn and Hüllermeier (2009) utilize ordered binary trees. Platt et al. (1999) construct a directed acyclic graph of base classifiers. Other authors present ordinal extensions of binary classifiers, such as linear classifiers (Cardoso and Pinto da Costa, 2007; Crammer and Singer, 2001).

In previous work, we showed that the susceptibility of ordinal classifier cascades could be used to reject wrong assumptions on the ordering of classes (Lattke et al., 2015). Explorative screens can utilize this property and return a set of candidate class orderings that are reflected by the feature space (Lausser et al., 2019).

In this work, we investigate the decision regions induced by ordinal classifiers. By analysing the neighborhood of the decision regions (Figure 1), we show that different partitions of a feature space exist that might be considered as ordinal. They can be classified according to the existence of disconnected decision regions, which can be required for handling unconsidered subcategories, and their neighborhoods. In this case there might be a local ordinal structure for each subcategory but no global one for the joint classes. Subcategories can occur at specific stages of an ordinal process leading to disconnected decision regions for the subsequent classes. An example might be a differentiation of cell types that is not considered in the labeling (e.g. labeling based on point in time) but it is expected to be seen in the feature representation. Inducing the possibility of disconnected decision regions within the ordinal classifier cascade allows a wider range of partitions to be accepted as ordinal.

However, this relaxation bears the risk of a specific type of false positive detection. We show that an ordinal classifier cascade might not detect (partial) repetitive structures within the feature space. We indicate this phenomenon in the screening experiments on artificial datasets with ordinal and repetitive class structures.

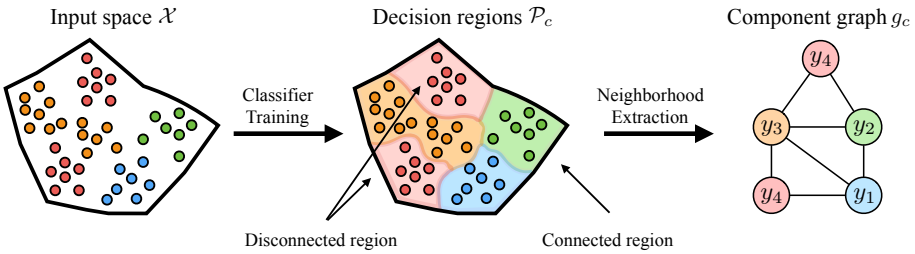


Figure 1: Construction of components graph g_c. In this work, we analyze the partition $\mathcal{P}_c$ of an feature space $\mathcal{X}$ induced by the training of a classifier c. By extracting the neighborhoods of the partitions we result in a component graph g_c. This graph can be analyzed to detect paths that are labeled according to an assumed class order.

The remaining article is organized as follows. Section 2 provides the underlying concepts and notation of the article. It especially gives a formal definition of ordinal classification (Section 2.1), our criteria for ordinal feature embeddings and finally the analyzed ordinal classifier cascades (Section 2.1.1). The experimental setup is given in Section 3. The corresponding results are provided in Section 4 and discussed in Section 5.

2 Methods

We will use the following notation throughout this article. An object will be represented as an n-dimensional feature vector $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})^T$ from a real-valued topological space $(\mathcal{X}, \omega)$ with $\mathcal{X} \subseteq \mathbb{R}^n$ and ω being the Euclidean topology. Each object is assumed to be categorizable into exactly one class y_i of a predefined set of classes $\mathcal{Y} = \{y_i\}_{i=1}^{|\mathcal{Y}|}$. The task of classification is to predict the correct class label of an object according to the available measurements. We distinguish between binary classification tasks ($|\mathcal{Y}| = 2$) and multi-class classification tasks ($|\mathcal{Y}| > 2$). A classifier is a function $c : \mathcal{X} \to \mathcal{Y}$. It is typically trained in a data-driven procedure l

$$l : \mathcal{C} \times \mathcal{T} \to c_{\mathcal{T}} \in \mathcal{C}. \tag{2}$$

Here, the symbol $\mathcal{C}$ denotes the concept or function class a classifier is chosen from. The symbol $\mathcal{T} = \left\{(\mathbf{x}_j, y_j)\right\}_{j=1}^{|\mathcal{T}|}$ denotes a set of labeled training examples

to which the classifier $c_{\mathcal{T}}$ was adapted. The subscript $_{\mathcal{T}}$ will be dropped, if the training set is clear from the context.

The generalization performance of a classifier c is evaluated on an independent set of labeled validation samples $\mathcal{V} = \left\{(\mathbf{x}'_j, y'_j)\right\}_{j=1}^{|\mathcal{V}|}$. In our study we focus on the class-wise sensitivities

$$sens(c, y) = \frac{1}{|\mathcal{V}_y|} \sum_{(\mathbf{x},y)\in\mathcal{V}_y} \mathbb{I}_{[c(x)=y]}, \tag{3}$$

where $\mathcal{V}_y = \{(\mathbf{x}', y')|(\mathbf{x}', y') \in \mathcal{V}, y' = y\}$.

Connected and Disconnected Decision Regions

We assume a classifier to assign a class label $y \in \mathcal{Y}$ to each $\mathbf{x} \in \mathcal{X}$. In this case, a classifier c constructs a partition $\mathcal{P}_c = \left\{\mathcal{D}_y\right\}_{y\in\mathcal{Y}}$ of decision regions $\mathcal{D}_y$ with $\mathcal{D}_y \cap \mathcal{D}_{y'} = \emptyset$ if $y \neq y'$ and $\mathcal{X} = \bigcup_{y\in\mathcal{Y}} \mathcal{D}_y$. Class label y is predicted by a classifier c if $\mathbf{x}$ lies in the corresponding decision region $\mathcal{D}_y$

$$c(\mathbf{x}) = y \quad \Longleftrightarrow \quad \mathbf{x} \in \mathcal{D}_y. \tag{4}$$

As an implication a classifier c is unable to predict class y, if $\mathcal{D}_y = \emptyset$. $\mathcal{D}_y$ can be further partitioned in terms of connected topological spaces and components (e.g. Massey (1967); Buskes and van Rooij (1997)). In the following, we use the notion of *path connectedness*.

Definition 1 (Path Connectedness) A topological space $\mathcal{D}$ is *path connected* if, for any two points $\mathbf{x}, \mathbf{x}' \in \mathcal{D}$ there is a continuous map

$$f : [0, 1] \longrightarrow \mathcal{D} \tag{5}$$

such that $f(0) = \mathbf{x}$ and $f(1) = \mathbf{x}'$. It allows the definition of (maximal) path connected components of a topological space.

Definition 2 (Path Connected Component) A subset $k \subseteq \mathcal{D}$ of a topological space $\mathcal{D}$ is a *path connected component* of $\mathcal{D}$, if k is path connected and there is no other path connected subset k' with $k \subset k' \subseteq \mathcal{D}$.

A component k is a maximal subset of $\mathcal{D}$ that fulfils path connectedness. As a direct consequence we get $k \neq \emptyset$ if $\mathcal{D} \neq \emptyset$. We can now provide the set of connected components $\mathcal{K}_y = \{k_i\}_{i=1}^{|\mathcal{K}_y|}$ for each decision region $\mathcal{D}_y$. As a consequence of Definition 2 the components of $\mathcal{K}_y$ are pairwise disjoint, their number $|\mathcal{K}_y|$ is minimal. $\mathcal{K}_y$ therefore again fulfils the properties of a partition

$$\mathcal{D}_y = \bigcup_{k \in \mathcal{K}_y} k \quad \text{and} \quad k \cap k' = \emptyset \text{ for all } k \neq k' \text{ with } k, k' \in \mathcal{K}_y. \tag{6}$$

A decision region $\mathcal{D}_y$ will be called *connected* if $|\mathcal{K}_y| = 1$ and *disconnected* otherwise ($|\mathcal{K}_y| > 1$). We will additionally use the notation $\mathcal{K}_c = \bigcup_{y \in \mathcal{Y}} \mathcal{K}_y$ to denote the set of all components of a classifier c. It will later on be used to define the graph of neighboring components (Definition 4). We will additionally use the notion of boundaries in order to define neighored components.

Definition 3 (Boundary) Let $\mathcal{D}$ be a topological space and $k \subseteq \mathcal{D}$. The *boundary* $\delta(k)$ of k is given be the set of points adherent to k and $\mathcal{D} \setminus k$.

Two components $k \in \mathcal{K}_y$ and $k' \in \mathcal{K}_{y'}$ share a common (decision) boundary if $\delta(k) \cap \delta(k') \neq \emptyset$.

2.1 Ordinal Classification

Ordinal classification is a multi-class classification task ($|\mathcal{Y}| > 2$). That is, we assume all semantic concepts to be pairwise distinct

$$\forall\, y, y' \in \mathcal{Y} : y \neq y'. \tag{7}$$

In ordinal classification we additionally assume a (total) semantic order of the class labels to be represented in the feature space

$$y_{(1)} \prec \ldots \prec y_{(|\mathcal{Y}|)}, \tag{8}$$

where $y_{(i)} \in \mathcal{Y}$ denotes the i-th class of the order. The symbol $\prec$ indicates that the ordering is only known (or assumed) for the underlying semantic concepts. Its reflection in the feature space is unknown and can not be guaranteed.

The assumed class order guides the design or the training of an ordinal classifier. For example, it can be utilized for defining the structure of an hierarchical classifier system (Ben-David, 1995). It can also be used for weighting the training samples of cost sensitive base classifiers (Lin and Li, 2012). Alternatively, specialised performance measures might be applied (Waegeman et al., 2008).

The decision regions of an ordinal classifier should finally reflect the proposed sequence of classes. Wrong assumptions on the class order should lead to a decreased classification performance. In a previous article (Lattke et al., 2015), we proposed to utilize this susceptibility of an ordinal classifier for a performance-based criterion on the ordinality of a dataset. Its minimal class-wise sensitivity evaluates a trained ordinal classifier c

$$\min_{y \in \mathcal{Y}} \quad sens(c, y). \tag{9}$$

By conducting classification experiments for all $|\mathcal{Y}|!$ possible class orderings, the influence of one specific class order can be judged concerning all other orderings (Lausser et al., 2019).

Here, we focus on the required structural properties of an ordinal classifier. The semantic order relationship (in the label space) fulfils the following characteristics which are required in the feature space:

$$\forall\, y \in \mathcal{Y} : \neg(y \prec y)\,, \qquad \text{[irreflexivity]} \tag{10}$$

$$\forall\, y, y' \in \mathcal{Y} : (y \prec y') \Longrightarrow \neg(y' \prec y)\,, \qquad \text{[asymmetry]} \tag{11}$$

$$\forall\, y, y', y'' \in \mathcal{Y} : (y \prec y') \wedge (y' \prec y'') \Longrightarrow (y \prec y'')\,. \qquad \text{[transitivity]} \tag{12}$$

These properties should again be verifiable by an ordinal classifier. Its decision regions should, therefore, fulfill the following minimal requirements:

1. All decision regions of an ordinal classifier should be non empty $\forall y : \mathcal{D}_y \neq \emptyset$.
2. The decision regions of two consecutive classes $\mathcal{D}_{y_{(i)}}$ and $\mathcal{D}_{y_{(i+1)}}$ should share a common decision boundary.

In order to provide concrete criteria for the second requirement we define the component graph g_c of a classifier c. For g_c we define the concept of *ordinal paths*, which allows us to formulate criteria based on the class labels and the neighborhood of all components $\mathcal{K}_c$.

Definition 4 (Component Graph) The *component graph* g_c of a classifier c is defined as an undirected simple graph $\mathcal{G}(\mathcal{K}_c, \mathcal{E}_c)$ with the set of vertices given by the components $\mathcal{K}_c$ and the set of edges $\mathcal{E}_c \subseteq \{(k_i, k_j) : (k_i, k_j) \in \mathcal{K}_c \times \mathcal{K}_c, i < j\}$ defined by the neighborhood

$$(k, k') \in \mathcal{E} \quad \Longleftrightarrow \quad \delta(k) \cap \delta(k') \setminus \bigcup_{k'' \in \mathcal{K}_c \setminus \{k, k'\}} \delta(k'') \neq \emptyset. \tag{13}$$

Definition 4 calls two components $k, k' \in \mathcal{K}_c$ neighboring, if they exclusively share a common decision boundary. That is, there is no other component $k'' \in \mathcal{K}_c \setminus \{k, k'\}$ that is adherent to the common decision boundary. In this case, the decision boundary is a (local) dichotomy between k and k'. Due to our definition of connected components, two neighboring components are guaranteed to be assigned to different classes.

In general, the graph g_c will comprise several paths, which can be analyzed individually. Hereby, each component can occur only once in each path and the order of class labels induced by the order of components can be inspected. This analysis connects the information of the feature space (components) with the label information. If each label is represented at least once, the definition of a class complete path is fulfilled (Definition 5). This criteria is a prerequisite for an ordinal path as the order in the label space is defined for all class labels.

Definition 5 (Class Complete Path) A path $(k_1, \ldots, k_p)$ in g_c is *class complete* (for $\mathcal{Y}$) if $\forall y \in \mathcal{Y}, \exists i : k_i \subseteq \mathcal{D}_y$.

One real life example of a non class complete path might be the skipping of a stage within a developmental process.

If there are disconnected decision regions, several nodes describe the same label and hence one class might be represented several times within a path. In this case, also patterns of class labels might be observed. If this pattern represents the expected order it refers to a repetitive class structure. As consequence a repetitive class structure consists of a path that is made up of several ordinal paths (Definition 6). If only parts of the ordinal path are repeated we call it partial repetitive class structure. In real life repetitive class structures might be seen if trends in a set of categorical data are analyzed.

Definition 6 (Ordinal Path) A path $(k_1, \ldots, k_{|\mathcal{Y}|})$ in g_c is *ordinal* (for $y_{(1)} \prec \ldots \prec y_{(|\mathcal{Y}|)}$) if $\forall i$: $k_i \subseteq \mathcal{D}_{y_{(i)}}$.

All ordinal paths in g_c are class complete. A path that is considered to be ordinal should additionally have the same order of labels as the assumed class order. As a consequence it should be of length $|\mathcal{Y}|$. They fulfill irreflexivity, asymmetry and transitivity. They can be used to define a hierarchy of at least four different partition types that could be identified as ordinal class structures. From less restrictive to more restrictive these criteria are described in the following. An illustration is given in Figure 2.

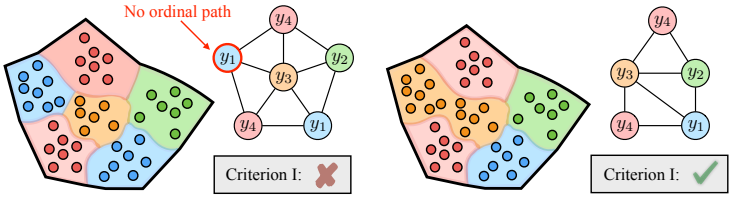
Criterion I: For each $i \in \{1, \ldots, |\mathcal{Y}|\}$ and for each component $k_i \in \mathcal{K}_{y_{(i)}}$ there exists an ordinal path $(k_1, \ldots, k_i, \ldots, k_{|\mathcal{Y}|}) \in g_c$ with $\forall j \neq i$: $k_j \in \mathcal{K}_{y_{(j)}}$.

Criterion III: Criterion I is fulfilled and one class is represented by a connected decision region.

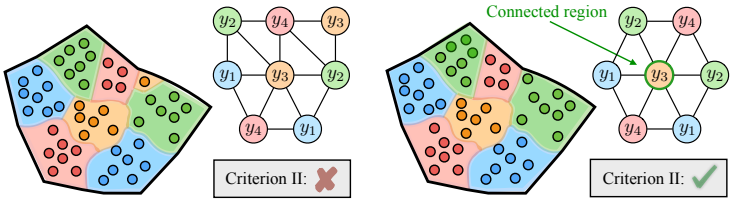
Criterion III: Criterion I is fulfilled and all classes are represented by connected decision regions.

Criterion IV: All class complete paths are ordinal paths (according to $y_{(1)} \prec \ldots \prec y_{(|\mathcal{Y}|)}$ or $y_{(|\mathcal{Y}|)} \prec \ldots \prec y_{(1)}$).

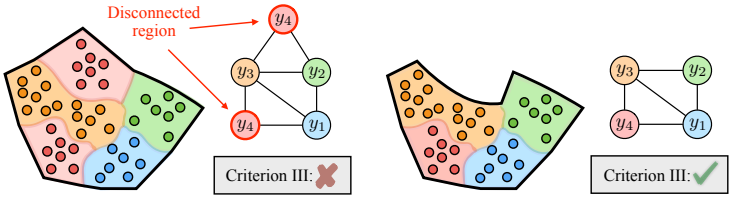
One aspect that differentiates these criteria is the restriction to connected decision regions (Figure 2). Whereas criterion I does not require any decision region to be connected and focuses on the general assumption that each component is an element of at least one ordinal path (Panel 2a), criterion II asks additionally for one connected decision region (Panel 2b). The more connected decision regions, the more restrictive the criterion becomes, leading to criterion III (Panel 2c), which assumes precisely as many decision regions as different class labels exist. The most restrictive criterion in our list is criterion IV (Panel 2d). To fulfill criterion IV, only consecutive classes are allowed to share a common decision boundary.



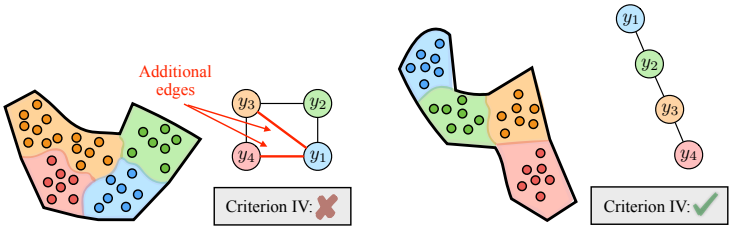
(a) Criterion I requires that each component to be part of an ordinal sequence. It can be seen that in the left representation this criterion is not fulfilled a as the blue component (marked with a red border) does not have a green component (there is no edge from this node to a green node) as neighbor. In the right representation for each component one ordinal path can be found.



(b) Criterion II requires additionally that at least one connected decision region exists. This is illustrated by the orange class, which has only one decision region if the criterion is fulfilled (right), but two regions in the counterexample (left).



(c) Criterion III requires all decision regions to be connected. In this example the red class shows a disconnected region in the counterexample (left), but each class has only one decision region in the positive example (right).



(d) Criterion IV requires that all paths are ordinal. In the left representation one can find a sequence $y_2 \prec y_3 \prec y_4 \prec y_1$, which is not the assumed class order and hence this requirement is not fulfilled. If each decision region shares a boundary only with its assumed neighboring region criterion IV is fulfilled (right).

Figure 2: Examples of different ordinal class topologies, their corresponding component graph and their fulfilment of criteria I - IV. Class labels are represented by coloured circles and are ordered by $y_1 \prec y_2 \prec y_3 \prec y_4$.

All criteria can occur in realistic scenarios. A representation according to criterion I might be the result of unconsidered subcategories (e.g. females and males) within the class definition. These subcategories might lead to disconnected decision regions that are localized differently in feature space. Nevertheless, the same ordinal process can be assumed in each subcategory. In the context of severe diseases (e.g. tumorigenesis) the characteristics of a disease might dominate the characteristics of the subcategories in the final stages and hence samples of these stage lay in the same decision region. This scenario would fulfill criterion II. A domination of this type might occur at several stages. If it is assumed for all stages one ends up at criterion III. Criterion IV corresponds to a reflection without any intersecting boundaries. As an example, this can be referred to a gradual shift of the features, as it might occur in subsets of features during tumorigenesis, or in dose-response measurements.

2.1.1 Ordinal Classifier Cascades

As an example for an ordinal classifier, we concentrate on ordinal classifier cascades (Frank and Hall, 2001). An ordinal classifier cascade can be seen as asking a sequence of experts, where each expert has its unique field of knowledge. An expert can decide that an object belongs to its field of knowledge (class) or pass the object to the next expert. The order of experts can influence the answer. A general scheme is shown in Figure 3.

Formally, a (full) ordinal classifier cascade $c(\mathbf{x})$ is a multi-class classifier scheme that combines an ensemble of base classifiers $\mathcal{B} = \left\{bc_{(1)}, \ldots, bc_{(|\mathcal{Y}|-1)}\right\}$ according to the predefined class order

$$c(\mathbf{x}) = \begin{cases} y_{(i)} & \text{if } bc_{(i)}(\mathbf{x}) = y_{(i)} \wedge \forall\, j < i : bc_{(j)}(\mathbf{x}) \neq y_{(j)} \\ y_{(|\mathcal{Y}|)} & \text{else.} \end{cases} \tag{14}$$

The i-th base classifier $bc_{(i)}$ is trained to distinguish between classes $y_{(i)}$ and $y_{(i+1)}$

$$bc_{(i)} : \mathcal{X} \longrightarrow \left\{y_{(i)}, y_{(i+1)}\right\}. \tag{15}$$

The cascade c generates an unambiguous partition of the overlapping decision regions of the base classifiers $bc_{(i)}$. Starting with the first classifier $bc_{(1)}$, the base classifiers are evaluated sequentially. The procedure stops when the i-th

classifier predicts class $y_{(i)}$. Otherwise, classifier $bc_{(i+1)}$ is evaluated. If the ordinal cascade predicts $c(\mathbf{x}) = y_{(i)}$, a preamble of $i-1$ base classifiers is evaluated negatively ($\forall\, j < i : bc_{(j)}(\mathbf{x}) \neq y_{(j)}$). In summary, predicting class $y_{(i)}$ requires a lower number of base classifiers than predicting $y_{(i+1)}$. The prediction of class $y_{(i)}$ is also faster, if we assume a constant prediction time of the base classifiers (Viola and Jones, 2004).

As the prediction of the class label $y_{(i)}$ and the i-th preamble size correspond one by one, the ordinal classifier cascade directly reflects the predefined semantic ordinal relationship on the fusion level. Nevertheless, the cascade depends on the input of its base classifiers.

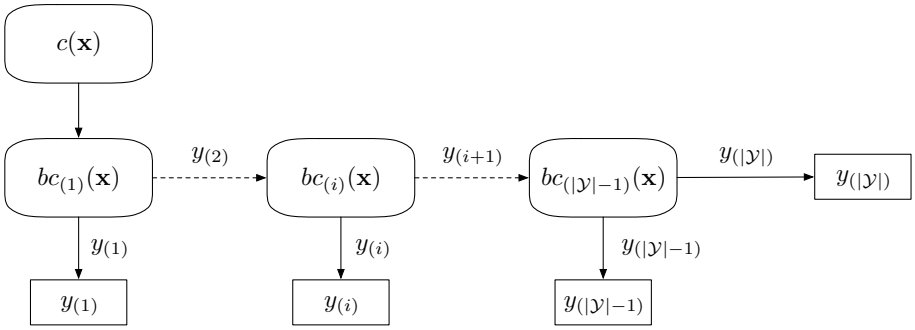


Figure 3: Scheme of an ordinal classifier cascade. The ordinal classifier cascade $c(\mathbf{x})$ consists of a sequence of binary base classifiers $bc_{(i)}(\mathbf{x})$. In this example an incremental order of class labels $y_{(i)}$ with $i = (1, \ldots, |\mathcal{Y}|)$ is assumed. This means that the i-th base classifier $bc_{(i)}(\mathbf{x})$ is trained to distinguish between class labels $y_{(i)}$ and $y_{(i+1)}$. If a base classifier predicts its first class $y_{(i)}$ for a given sample $\mathbf{x}$ the cascade stops and the sample is labelled as $y_{(i)}$, otherwise the sample is evaluated by the next base classifier in the sequence. If no base classifier predicts its first class the sample is labelled as $y_{(|\mathcal{Y}|)}$.

2.1.2 On the Choice of Base Classifiers

The ordinal classifier cascade is a late aggregation fusion architecture. That means it has only access to the predictions of its base classifiers; it does not receive any further information about the feature space. The ensembles of ordinal multi-class architectures consist of binary (non-ordinal) base classifiers; they will typically not report violations of ordinal assumptions. Moreover, the choice and design of a base classifier can itself harm the ordinal assumptions.

An example would be base classifiers that allow disconnected decision regions. The previously defined criteria impose different assumptions on the connectivity of the decision regions. Criterion I is still fulfilled even if there are no connected decision regions and also criterion II allows for disconnected decision regions. In contrast to that, criterion III and IV require connected decision regions. Using base classifiers that have connected decision regions ordinal patterns according to criterion I and II that do not fulfill III cannot be found. To detect those one needs the flexibility of base classifiers that can divide the input space $\mathcal{X}$ into decision regions that are disconnected. Those classifiers, however, do not guarantee that criterion I is fulfilled.

3 Experiments

In our experiments, we evaluate ordinal classifier cascades based on binary classifiers with connected and disconnected decision regions. Cascades are trained and tested for all possible $|\mathcal{Y}|!$ orderings of the class label. If not stated otherwise the base classifiers $bc_{(i)}$ are trained in a pairwise manner (Lattke et al., 2015). That is each base classifier is trained on the samples of classes $y_{(i)}$ and $y_{(i+1)}$.

For each experiment, the class-wise sensitivities are reported. The evaluation is designed as a 10×10 cross-validation experiment (Japkowicz and Shah, 2011) and is performed with the TunePareto R-package (Müssel et al., 2012).

k-nearest Neighbors (*k*-NN)

As an example for a classifier with disconnected decision boundaries we analyze the well known k-Nearest Neighbor classifier (k-NN) (Fix and Hodges, 1951). The k-NN is a member of the family of prototype-based classifiers that utilize a set of labeled prototypes $O = \left\{(\mathbf{x}_j, y_j)\right\}_{j=1}^{|O|}$ for the prediction of the class label of a validation sample $\mathbf{v}$

$$bc(\mathbf{v}) = \underset{y\in\mathcal{Y}}{\operatorname{argmax}} \left|\{(\mathbf{x}, y) \in \mathrm{NN}_k(\mathbf{v}, O)\}\right|, \tag{16}$$

where NN_k is the k nearest neighborhood of $\mathbf{v}$ in O

$$\mathrm{NN}_k = \left\{(\mathbf{x}, y) \in O | \mathrm{rk}_{\mathcal{D}_\mathbf{v}}\left(d(\mathbf{v}, \mathbf{x}) \leq k\right)\right\} \tag{17}$$

and $\mathrm{rk}_{\mathcal{D}_\mathbf{v}}$ is the rank function on the pairwise distance d of $\mathbf{v}$ and the elements of O

$$D_\mathbf{v} = \{d(\mathbf{v}, \mathbf{x}) | (\mathbf{x}, y) \in O\}\,. \tag{18}$$

In the standard version of k-NN, $O = \mathcal{T}$ and the Euclidian distance is used. In general each k neighborhood results in an individual decision region. The connection of the decision regions can not be guaranteed. We utilize $k = 3$ neighbors in the following.

Linear Support Vector Machines (SVM)

As a base classifier with connected decision regions, we utilize the linear support vector machine (SVM) (Vapnik, 1998). It can be seen as a linear classifier of type

$$bc(\mathbf{x}) = \begin{cases} y_1 & \text{if} \quad \mathbf{w}^T\mathbf{x} \geq t \\ y_2 & \text{else} \end{cases}\,, \tag{19}$$

which is trained to maximize the margin between two classes y_1 and y_2. In this context $\mathbf{w} \in \mathbb{R}^n$ can be seen as the multi-dimensional angle of a hyperplane and $t \in \mathbb{R}$ as its distance to the origin. The training of the linear SVM is a constrained minimisation of the following objective:

$$\min_{\mathbf{w},t,\xi} \quad \|\mathbf{w}\|_2^2 + C\sum_{j=1}^{|\mathcal{T}|}\xi_j \tag{20}$$

$$\text{s.t.} \quad \forall_{j=1}^{|\mathcal{T}|} : y_j(\mathbf{w}^T\mathbf{x}_j) - t \geq 1 - \xi_j \tag{21}$$

$$\forall_{j=1}^{|\mathcal{T}|} : \xi_j \geq 0, \tag{22}$$

$\mathcal{T} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{|\mathcal{T}|}$ with label space $\mathcal{Y} = \{-1, +1\}$. A positive slack variable ξ_j describes the wrong classification of sample $\mathbf{x}_j$ and gives its distance to the margin. Parameter C determines the ratio between strict separation and misclassified samples.

In its two-class version, the linear SVM splits the input space into two decision regions. Since these decision regions are connected, we assume it to be suitable for scenarios that fulfill criterion III.

Parallel Decision Boundary Support Vector Machines (par-SVM)

In a second experiment, we further constrained the ordinal cascade of linear SVMs to operate on a common orientation vector ($\mathbf{w}$), which results in parallel decision boundaries for all base classifiers. The decision boundaries will only differ in their thresholds $\mathbf{t} = (t^{(i)})_{i=1}^{|\mathcal{Y}|-1}$:

$$\min_{\mathbf{w},\mathbf{t},\Xi} \quad \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{|\mathcal{Y}|-1} \sum_{j=1}^{|\mathcal{T}|} \xi^{(i,j)} \tag{23}$$

$$\text{s.t.} \quad \forall_{i=1}^{|\mathcal{Y}|-1} \forall_{j=1}^{|\mathcal{T}|} : \tilde{y}_{i,j}(\mathbf{w}^T \mathbf{x}_j) - t^{(i)} \geq 1 - \xi^{(i,j)} \tag{24}$$

$$\forall_{i=1}^{|\mathcal{Y}|-1} \forall_{j=1}^{|\mathcal{T}|} : \xi^{(i,j)} \geq 0. \tag{25}$$

In this context $\Xi = (\xi^{(i,j)})_{i \in \{1,\ldots,|\mathcal{Y}|-1\}, j \in \{1,\ldots,|\mathcal{T}|\}}$ and

$$\tilde{y}_{i,j} = \begin{cases} -1 & \text{if} \quad y_j \in \{y_{(1)}, \ldots, y_{(i)}\} \\ +1 & \text{otherwise} \end{cases} . \tag{26}$$

This classifier is not trained in a pairwise manner but on all class labels. We assume that it is able to distinguish criterion III and criterion IV, as it prevents decision borders from intersecting each other.

3.1 Datasets

We evaluate the ordinal classifier cascades on ten artificial datasets $d_1, ..., d_{10}$ (Figure 4 and Figure 5). Their characteristics, especially the fulfilled ordinality criteria, are listed in Table 1. The first six datasets $d_1, ..., d_6$ are one dimensional datasets with $\mathcal{X} \in \mathbb{R}$, $n = 1$. The remaining datasets $d_7, ..., d_{10}$ comprise two dimensional patterns ($\mathcal{X} \in \mathbb{R}^2$, $n = 2$). The used patterns can also be embedded in higher dimensions. This up-scaling will not influence the corresponding component graph. All datasets comprise four classes $\mathcal{Y} = \{y_1, \ldots, y_4\}$ and are constructed according to the same principle.

For each class $y \in \mathcal{Y}$ a set of centroids $\mathcal{Z}_y = \left\{\mathbf{z}_{y,i}\right\}_{i=1}^{|\mathcal{Z}_y|}$, $\mathbf{z}_{y,i} \in \mathbb{N}^n$ is chosen. For each centroid $\mathbf{z}_y \in \mathcal{Z}_y$ a set of 50 samples is drawn according to a Gaussian distribution

$$\mathbf{x} \sim \mathcal{N}(\mathbf{z}_y, \sigma \mathbf{I}^n). \tag{27}$$

Here, $\mathbf{I}^n$ denotes a n-dimensional unit matrix and $\sigma \in \mathbb{R}$. In our experiments, $\sigma = 0.3$ was chosen. The datasets therefore comprise $m_y = 50|\mathcal{Z}_y|$ samples for class y. All datasets are designed to represent the ordinal class structure

$$y_1 \prec y_2 \prec y_3 \prec y_4. \tag{28}$$

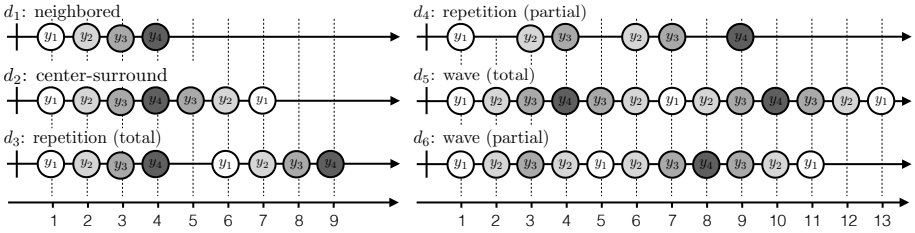


Figure 4: Schematic graphs of one-dimensional datasets $d_1, \ldots, d_6$. Each dataset consists of four classes $y_1, \ldots, y_4$. A single circle denotes a cluster of 50 samples of the corresponding class. The coordinate system defines the location of the class centroids.

The exact position of the centroids can either be extracted from Figure 4 and Figure 5 or taken from the Supplementary Table A.1. Datasets $d_1, \ldots, d_6$ are one-dimensional datasets (Figure 4). Dataset d_1 comprises one centroid per class. The order of the classes corresponds to the size of the centroids x-values. In dataset d_2 the (single) centroid $\mathbf{z}_{y_4} = 4$ of class y_4 is surrounded by pairs of centroids of the classes $y_1 - y_3$.

The centroids of datasets $d_3, \ldots, d_6$ are organized in repetitive structures. In dataset d_3, the whole sequence of centroids of dataset d_1 is duplicated and shifted to a distinct position. In dataset d_4 only the subsequence of classes y_2, y_3 is duplicated and placed twice between classes y_1 and y_4. Dataset d_5 can be seen as a repetitive version of dataset d_2 leading to a wave-like pattern in the class neighborhoods. The same principle is used for dataset d_6. In contrast to d_5, only the classes $y_1 - y_3$ are used for the second wave.

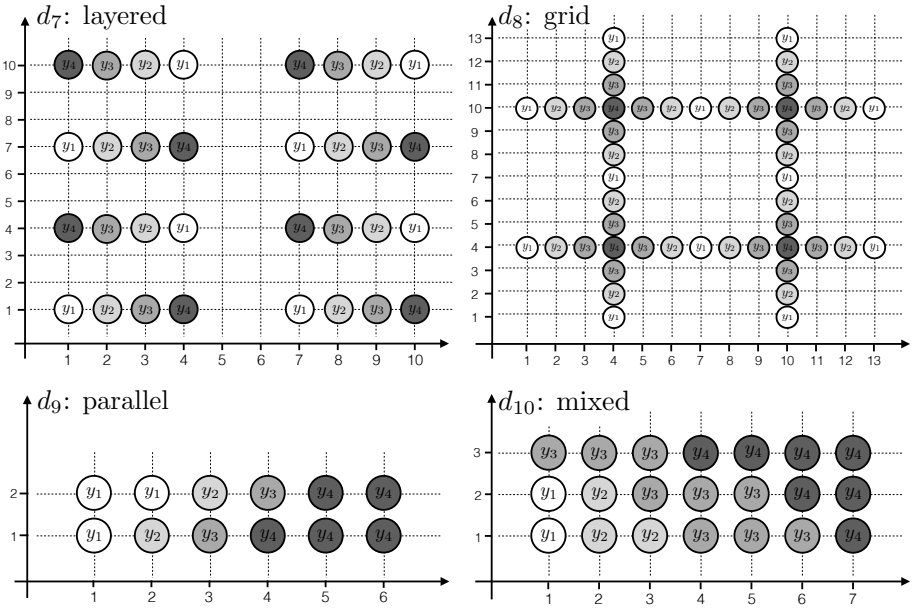


Figure 5: Schematic graphs of two-dimensional datasets d_7 - d_{10}. Each dataset consists of four classes $y_1, \ldots, y_4$. A single circle denotes a cluster of 50 samples of the corresponding class. The coordinate system defines the location of the class centroids.

Table 1: Main characteristics of the analyzed datasets. The criteria fulfilled, the numbers of samples m, samples per class m_y and the features n (dimensionality) are reported. All datasets comprise samples of $|\mathcal{Y}| = 4$ classes.

Name	**Criterion**	**m**	**$\mathbf{m_y}$**	**n**
d_1: neighboring	I, II, III, IV	200	50, 50, 50, 50	1
d_2: center-surround	I, II	350	100, 100, 100, 50	1
d_3: repetition (total)	I	400	100, 100, 100, 100	1
d_4: repetition (partial)	- - -	300	50, 100, 100, 50	1
d_5: wave (total)	I	650	150, 200, 200, 100	1
d_6: wave (partial)	- - -	550	150, 200, 150, 50	1
d_7: layered	I	550	400, 400, 400, 400	2
d_8: grid	I	2400	600, 800, 800, 200	2
d_9: parallel	I, II, III, IV	600	150, 100, 100, 250	2
d_{10}: mixed	I, II, III	1050	100, 150, 450, 350	2

Examples of two-dimensional datasets are shown in Figure 5. Dataset d_7 is a layered version of dataset d_2. In each layer, the class order of the prototypes is reversed. Dataset d_8 shows a two-dimensional grid of wave structures. Dataset d_9 shows a parallel striped pattern whereas in dataset d_{10} the parallelism is disturbed.

Per construction the datasets are assumed to meet several of the previously defined criteria. The datasets and their criteria are listed in Table 1. Datasets that do not fulfill criterion I are considered as non ordinal. Repetitive and hence local ordinal structures, exists if for multiple classes two centroids that are not neighboring by construction exist and if these duplicates are of the same order.

4 Results

The results of the 10×10 CV experiments are given in Table 2 (3-NN), Table 3 (SVM) and Table 4 (par-SVM). For each dataset, all cascades that achieve a minimal class-wise sensitivity higher than $t = 60.0\,\%$ in the 10×10 CV are listed. An overview on the complete confusion tables is given in Appendix A.2. Over all experiments, only the assumed class order and its reverse passed this limit. The ordinal cascades based on 3-NN classifiers achieved minimal class-wise sensitivities of at least 66.9 % on all datasets. The cascades based on linear SVMs detected class orders only for the 1D dataset d_1 (neighboring) and the 2D datasets d_9 (parallel) and d_{10} (mixed). For all other datasets, all candidate cascades are rejected. Additionally, the SVM with parallel decision boundaries was tested. It suggests the assumed class order and its reverse for d_1 (neighboring) and d_9 (parallel) whereas it fails to detect any ordinal pattern in the other datasets.

Table 2: Results for ordinal classifier cascades based on independent 3-NN classifiers. For each dataset, the cascades with a minimal class-wise sensitivity of at least $t = 60.0\,\%$ are reported.

			$\mathbf{sens}(\mathbf{y})$			
ID	**Name**	**Class Order**	$\mathbf{y}_{(1)}$	$\mathbf{y}_{(2)}$	$\mathbf{y}_{(3)}$	$\mathbf{y}_{(4)}$
d_1	neighboring	$y_1 \prec y_2 \prec y_3 \prec y_4$	98.0 %	94.2 %	91.8 %	96.2 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	98.0 %	94.2 %	91.8 %	96.2 %
		others		$\min_y sens(y) < t$		
d_2	center-	$y_1 \prec y_2 \prec y_3 \prec y_4$	91.5 %	86.5 %	84.7 %	89.8 %
	surround	$y_4 \prec y_3 \prec y_2 \prec y_1$	91.5 %	86.5 %	84.7 %	89.8 %
		others		$\min_y sens(y) < t$		
d_3	repetition	$y_1 \prec y_2 \prec y_3 \prec y_4$	95.7 %	90.8 %	94.1 %	73.2 %
	(total)	$y_4 \prec y_3 \prec y_2 \prec y_1$	66.9 %	90.8 %	94.1 %	95.3 %
		others		$\min_y sens(y) < t$		
d_4	repetition	$y_1 \prec y_2 \prec y_3 \prec y_4$	100.0 %	93.0 %	94.4 %	100.0 %
	(partial)	$y_4 \prec y_3 \prec y_2 \prec y_1$	100.0 %	93.0 %	94.4 %	100.0 %
		others		$\min_y sens(y) < t$		
d_5	wave	$y_1 \prec y_2 \prec y_3 \prec y_4$	92.1 %	88.1 %	85.5 %	89.6 %
	(total)	$y_4 \prec y_3 \prec y_2 \prec y_1$	92.1 %	88.1 %	85.5 %	89.6 %
		others		$\min_y sens(y) < t$		
d_6	wave	$y_1 \prec y_2 \prec y_3 \prec y_4$	96.5 %	88.5 %	87.3 %	88.2 %
	(partial)	$y_4 \prec y_3 \prec y_2 \prec y_1$	96.5 %	88.5 %	87.3 %	88.2 %
		others		$\min_y sens(y) < t$		
d_7	layered	$y_1 \prec y_2 \prec y_3 \prec y_4$	92.5 %	88.7 %	88.6 %	89.4 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	88.4 %	88.7 %	88.6 %	92.5 %
		others		$\min_y sens(y) < t$		
d_8	grid	$y_1 \prec y_2 \prec y_3 \prec y_4$	91.5 %	89.5 %	89.2 %	81.3 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	91.5 %	89.5 %	89.2 %	81.3 %
		others		$\min_y sens(y) < t$		
d_9	parallel	$y_1 \prec y_2 \prec y_3 \prec y_4$	93.7 %	83.7 %	82.6 %	96.5 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	93.7 %	83.7 %	82.6 %	96.4 %
		others		$\min_y sens(y) < t$		
d_{10}	mixed	$y_1 \prec y_2 \prec y_3 \prec y_4$	92.9 %	85.9 %	82.1 %	93.3 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	71.6 %	85.9 %	92.8 %	93.3 %
		others		$\min_y sens(y) < t$		

Table 3: Results for ordinal classifier cascades based on independent SVM classifiers. For each dataset, the cascades with a minimal class-wise sensitivity of at least $t = 60.0\,\%$ are reported.

				$\mathbf{sens(y)}$		
ID	**Name**	**Class Order**	$\mathbf{y_{(1)}}$	$\mathbf{y_{(2)}}$	$\mathbf{y_{(3)}}$	$\mathbf{y_{(4)}}$
d_1	neighboring	$y_1 \prec y_2 \prec y_3 \prec y_4$	99.2 %	96.0 %	88.2 %	98.4 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	99.2 %	96.0 %	88.2 %	98.4 %
		others		$\min_y sens(y) < t$		
$d_2 - d_8$	Other datasets	all cascades		$\min_y sens(y) < t$		
d_9	parallel	$y_1 \prec y_2 \prec y_3 \prec y_4$	93.0 %	84.6 %	84.1 %	96.0 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	93.0 %	84.6 %	84.1 %	96.0 %
		others		$\min_y sens(y) < t$		
d_{10}	mixed	$y_1 \prec y_2 \prec y_3 \prec y_4$	93.8 %	76.1 %	81.4 %	89.1 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	93.8 %	76.1 %	89.3 %	89.1 %
		others		$\min_y sens(y) < t$		

Table 4: Results for ordinal classifier cascades based on parallel SVM classifier. For each dataset, the cascades with a minimal class-wise sensitivity of at least $t = 60.0\,\%$ are reported.

				$\mathbf{sens(y)}$		
ID	**Name**	**Class Order**	$\mathbf{y_{(1)}}$	$\mathbf{y_{(2)}}$	$\mathbf{y_{(3)}}$	$\mathbf{y_{(4)}}$
d_1	neighboring	$y_1 \prec y_2 \prec y_3 \prec y_4$	99.0 %	96.0 %	84.4 %	100 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	99.0 %	96.0 %	84.4 %	100 %
		others		$\min_y sens(y) < t$		
$d_2 - d_8$	Other datasets	all cascades		$\min_y sens(y) < t$		
d_9	parallel	$y_1 \prec y_2 \prec y_3 \prec y_4$	93.1 %	84.3 %	83.4 %	96.4 %
		$y_4 \prec y_3 \prec y_2 \prec y_1$	93.1 %	84.3 %	83.4 %	96.4 %
		others		$\min_y sens(y) < t$		
d_{10}	mixed	all cascades		$\min_y sens(y) < t$		

5 Discussion and Conclusion

In this work, we addressed the question of different types of ordinal reflection in feature space. While ordinal representations that fulfill at least criterion III reflect a (semantic) class order on a global level, repetitive representations reflect this order in multiple local structures. At the border of two local structures, the semantic ordinal relationship might not be fulfilled. Nevertheless, each local structure might reflect on its own the semantic ordinal relationship, and hence one might still conclude that an ordinal representation (of a less strict criterion) is also present in the feature space. If the ordinal reflection is neither given on the local nor on the global level, we do not consider the representation as ordinal. As a consequence partial repetitive structures are not viewed as an ordinal representation here.

Our experiments show that ordinal classifier cascades that operate on base classifiers with disconnected decision regions can neither distinguish between ordinal representation on a global and local level, nor between total and partial repetitions. Interestingly, these cascades do not loose their ability to detect a predefined semantic class order. The same class orderings were detected and rejected for ordinal and repetitive structures.

Whereas the ability not to differentiate between the local and global level allows for detecting data representations according to criterion I and II, not differentiating between partial and total repetitions leads to the detection of ordinality in feature representations that do not fulfill criterion I and are hence not considered as ordinal by us.

Besides the differentiation between disconnected and connected decision regions one could think of other criteria that allow a more fine-granular hierarchy of ordinal class structures. The criterion of connected decision regions (criterion III) can be even further restricted by the additional requirement of non intersecting decision boundaries (criterion IV). By constraining the cascade of linear SVMs (criterion III) to parallel decision boundaries (criterion IV), we show that not all datasets that allow a correct detection of a global ordinality (criterion III) also allow a detection according to criterion IV.

As a consequence one can conclude that the choice of base classifier defines the set of possible ordinal patterns that can be detected. The higher the flexibility of a base classifier, the more patterns are possi-

ble, with the risk that ordinality is detected in data representations that might not be considered as ordinal anymore.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°602783, the German Research Foundation (DFG, SFB 1074 project Z1 and GRK 2254 HEIST), and the Federal Ministry of Education and Research (BMBF, Gerontosys II, Forschungskern SyStaR, ID 0315894A and e:Med, SYMBOL-HF, ID 01ZX1407A , CONFIRM, ID 01ZX1708C) all to HAK.

References

Ben-David A (1995) Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms. Machine Learning 19(1):29–43. DOI: 10.1023/A:1022655006810.

Buskes G, van Rooij A (1997) Topological Spaces: From Distance to Neighborhood, 1st edn. Undergraduate Texts in Mathematics, Springer, New York (USA). DOI: 10.1007/978-1-4612-0665-1.

Cardoso J, Pinto da Costa J (2007) Learning to Classify Ordinal Data: The Data Replication Method. Journal of Machine Learning Research 8:1393–1429. URL: `http://www.jmlr.org/papers/v8/cardoso07a.html`.

Crammer K, Singer Y (2001) Pranking with Ranking. In: Dietterich T, Becker S, Ghahramani Z (eds.), Neural Information Processing Systems (NIPS2001), MIT Press, Cambridge (USA), Advances in Neural Information Processing Systems, Vol. 14, pp. 641–647.

Fix E, Hodges JL (1951) Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties. Tech. Rep., USAF School of Aviation Medicine, Randolf Field (USA), pp. 261–279. Project 21-49-004, Report Number 4.

Frank E, Hall M (2001) A Simple Approach to Ordinal Classification. In: De Raedt L, Flach P (eds.), European Conference on Machine Learning (ECML2001), Springer, Berlin, Heidelberg (Germany), Lecture Notes in Computer Science, Vol. 2167, pp. 145–156. DOI: 10.1007/3-540-44795-4_13.

Hruban RH, Adsay NV, Albores-Saavedra J, Compton C, Garrett ES, Goodman SN, Kern SE, Klimstra DS, Klöppel G, Longnecker DS, Lüttges J, Offerhaus GJ (2001) Pancreatic Intraepithelial Neoplasia: A New Nomenclature and Classification System for Pancreatic Duct Lesions. The American Journal of Surgical Pathology 25(5):579–586. DOI: 10.1097/00000478-200105000-00003.

Hühn J, Hüllermeier E (2009) Is an Ordinal Class Structure Useful in Classifier Learning? Journal of Data Mining, Modelling and Management 1(1):45–67, Wang J (ed.). DOI: 10.1504/IJDMMM.2008.022537.

Japkowicz N, Shah M (2011) Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press, New York (USA). DOI: 10.1017/CBO9780511921803.

Lattke R, Lausser L, Müssel C, Kestler HA (2015) Detecting Ordinal Class Structures. In: Schwenker F, Roli F, Kittler J (eds.), Multiple Classifier Systems (MCS2015), Springer Verlag, Cham (Switzerland), Lecture Notes in Computer Science, Vol. 9132, pp. 100–111. DOI: 10.1007/978-3-319-20248-8_9.

Lausser L, Müssel C, Kestler HA (2013) Measuring and Visualizing the Stability of Biomarker Selection Techniques. Computational Statistics 28(1):51–65. DOI: 10.1007/s00180-011-0284-y.

Lausser L, Schmid F, Platzer M, Sillanpää MJ, Kestler HA (2014) Semantic Multi-classifier Systems for the Analysis of Gene Expression Profiles. Archives of Data Science, Series A 1(1):157–176. DOI: 10.5445/KSP/1000058747/09.

Lausser L, Szekely R, Schirra LR, Kestler HA (2018) The Influence of Multi-class Feature Selection on the Prediction of Diagnostic Phenotypes. Neural Processing Letters 48:863–880. DOI: 10.1007/s11063-017-9706-3.

Lausser L, Schäfer LM, Schirra LR, Szekely R, Schmid F, Kestler HA (2019) Assessing Phenotype Order in Molecular Data. Scientific Reports 9(1):11746. DOI: 10.1038/s41598-019-48150-z.

Lin HT, Li L (2012) Reduction from Cost-Sensitive Ordinal Ranking to Weighted Binary Classification. Neural Computation 24(5):1329–1367. DOI: 10.1162/NECO_a_00265.

Massey W (1967) Algebraic Topology : An Introduction, Graduate Texts in Mathematics, Vol. 56. Springer, New York (USA). ISBN: 978-0-387902-71-5.

Müssel C, Lausser L, Maucher M, Kestler HA (2012) Multi-Objective Parameter Selection for Classifiers. Journal of Statistical Software 46(5):1–27. DOI: 10.18637/jss.v046.i05.

Platt JC, Shawe-Taylor J, Cristianini N (1999) Large Margin DAG's for Multiclass Classification. In: Solla SA, Leen TK, Müller K (eds.), Proceedings of the 12th International Conference on Neural Information Processing Systems: Mini-Symposium on Causality in Time Series, MIT Press, Cambridge (USA), Advances in Neural Information Processing Systems, Vol. 12, pp. 547–553.

Rivest RL (1987) Learning Decision Lists. Machine Learning 2(3):229–246. DOI: 10.1007/BF00058680.

Schirra LR, Lausser L, Kestler HA (2016) Selection Stability as a Means of Biomarker Discovery in Classification. In: Wilhelm AFX, Kestler HA (eds.), Analysis of Large and Complex Data, Springer, Cham (Switzerland), pp. 79–89. DOI: 10.1007/978-3-319-25226-1_7.

Taudien S, Lausser L, Giamarellos-Bourboulis EJ, Sponholz C, F. S, Felder M, Schirra LR, Schmid F, Gogos C, S. G, Petersen BS, Franke A, Lieb W, Huse K, Zipfel PF, Kurzai O, Moepps B, Gierschik P, Bauer M, Scherag A, Kestler HA, Platzer M (2016) Genetic Factors of the Disease Course After Sepsis: Rare Deleterious Variants Are Predictive. EBioMedicine 12:227–238. DOI: 10.1016/j.ebiom.2016.08.037.

Vapnik VN (1998) Statistical Learning Theory, Adaptive and Learning Systems for Signal Processing, Communications, and Control, Vol. 2. John Wiley & Sons, New York (USA). ISBN: 978-0-471030-03-4.

Viola P, Jones M (2004) Robust Real-Time Face Detection. International Journal of Computer Vision 57(2):137–154. DOI: 10.1023/B:VISI.0000013087.49260.fb.

Waegeman W, Baets BD, Boullart L (2008) ROC Analysis in Ordinal Regression Learning. Pattern Recognition Letters 29(1):1–9. DOI: 10.1016/j.patrec.2007.07.019.

Weinberg R (2013) The Biology of Cancer, 2nd edn. Garland Publishing Inc. ISBN: 978-1-317963-46-2.

Appendix

A.1 Class Centroids of Artificial Datasets

Table A.1 provides the class centroids used for the construction of the artificial datasets.

Table A.1: Class centroids of artificial datasets (1/2).

ID	Name	Classes	Centroids
d_1	neighboring	$y_1, \ldots, y_4$	$\mathbf{z}_{y_i} = i$
d_2	center-surround	$y_1, \ldots, y_3$	$\mathbf{z}_{y_i,1} = \mathbf{z}_{y_4} - (4-i)$ $\mathbf{z}_{y_i,2} = \mathbf{z}_{y_4} - (4-i)$
		y_4	$\mathbf{z}_{y_4} = 4$
d_3	repetition (total)	$y_1, \ldots, y_4$	$\mathbf{z}_{y_i,1} = i$
		$y_1, \ldots, y_4$	$\mathbf{z}_{y_i,2} = i + 6$
d_4	repetition (partial)	y_1	$\mathbf{z}_{y_1} = 1$
		y_2, y_3	$\mathbf{z}_{y_i,1} = i + 1$ $\mathbf{z}_{y_i,3} = i + 4$
		y_4	$\mathbf{z}_{y_4} = 9$
d_5	wave (total)	$y_1, \ldots, y_4$	$\mathbf{z}_{y_i,1} = i$ $\mathbf{z}_{y_i,2} = i + 6$
		$y_1, \ldots, y_3$	$\mathbf{z}_{y_i,3} = 14 - i$
		y_2, y_3	$\mathbf{z}_{y_i,4} = 8 - i$
d_6	wave (partial)	$y_1, \ldots, y_4$	$\mathbf{z}_{y_i,1} = i + 4$
		$y_1, \ldots, y_3$	$\mathbf{z}_{y_i,2} = i$
		$y_1, \ldots, y_3$	$\mathbf{z}_{y_i,3} = 12 - i$
		y_2	$\mathbf{z}_{y_2,4} = 4$
d_7	layered	$y_1, \ldots, y_4$	$\mathbf{z}_{y_i,1} = (i, 1)^T$ $\mathbf{z}_{y_i,2} = \mathbf{z}_{y_i,1} + (6, 0)^T$ $\mathbf{z}_{y_i,3} = \mathbf{z}_{y_i,1} + (0, 6)^T$ $\mathbf{z}_{y_i,4} = \mathbf{z}_{y_i,1} + (6, 6)^T$ $\mathbf{z}_{y_i,5} = (5 - i, 4)^T$ $\mathbf{z}_{y_i,6} = \mathbf{z}_{y_i,5} + (6, 0)^T$ $\mathbf{z}_{y_i,7} = \mathbf{z}_{y_i,5} + (0, 6)^T$ $\mathbf{z}_{y_i,8} = \mathbf{z}_{y_i,5} + (6, 6)^T$

Table A.1: Class centroids of artificial datasets (2/2).

ID	Name	Classes	Centroids
d_8	grid	$y_1, \ldots, y_4$	$\mathbf{z}_{y_i,1} = (i, 4)^T$
			$\mathbf{z}_{y_i,2} = \mathbf{z}_{y_i,1} + (5, 0)^T$
			$\mathbf{z}_{y_i,3} = \mathbf{z}_{y_i,1} + (0, 6)^T$
			$\mathbf{z}_{y_i,4} = \mathbf{z}_{y_i,1} + (5, 6)^T$
		$y_1, \ldots, y_3$	$\mathbf{z}_{y_i,5} = (4, i)^T$
			$\mathbf{z}_{y_i,6} = \mathbf{z}_{y_i,5} + (6, 0)^T$
			$\mathbf{z}_{y_i,7} = \mathbf{z}_{y_i,5} + (0, 6)^T$
			$\mathbf{z}_{y_i,8} = \mathbf{z}_{y_i,5} + (6, 6)^T$
			$\mathbf{z}_{y_i,9} = (4, 14 - i)^T$
			$\mathbf{z}_{y_i,10} = \mathbf{z}_{y_i,9} + (6, 0)^T$
			$\mathbf{z}_{y_i,11} = (14 - i, 4)^T$
			$\mathbf{z}_{y_i,12} = \mathbf{z}_{y_i,11} + (0, 6)^T$
		y_2, y_3	$\mathbf{z}_{y_i,13} = (8 - i, 4)^T$
			$\mathbf{z}_{y_i,14} = \mathbf{z}_{y_i,12} + (0, 6)^T$
			$\mathbf{z}_{y_i,15} = (4, 8 - i)^T$
			$\mathbf{z}_{y_i,16} = \mathbf{z}_{y_i,15} + (6, 0)^T$
d_9	parallel	$y_1, \ldots, y_4$	$\mathbf{z}_{y_i,1} = (i, 1)^T$
			$\mathbf{z}_{y_i,2} = \mathbf{z}_{y_i,1} + (1, 1)^T$
		y_1	$\mathbf{z}_{y_i,3} = (i, 2)^T$
		y_4	$\mathbf{z}_{y_i,4} = (5, 1)^T$
			$\mathbf{z}_{y_i,5} = (6, 1)^T, (6, 2)^T$
d_{10}	mixed	y_1, y_2	$\mathbf{z}_{y_i,1} = (i, 1)^T$
			$\mathbf{z}_{y_i,2} = \mathbf{z}_{y_i,1} + (0, 1)^T$
		y_2	$\mathbf{z}_{y_i,3} = (3, 1)^T$
		y_3	$\mathbf{z}_{y_i,4} = ([1:3], 3)^T$
			$\mathbf{z}_{y_i,5} = \mathbf{z}_{y_i,4} + (2, -1)^T$
			$\mathbf{z}_{y_i,6} = \mathbf{z}_{y_i,4} + (3, -2)^T$
		y_4	$\mathbf{z}_{y_i,6} = (4, 3)^T, (5, 3)^T$
			$\mathbf{z}_{y_i,7} = \mathbf{z}_{y_i,6} + (2, 0)^T$
			$\mathbf{z}_{y_i,8} = \mathbf{z}_{y_i,6} + (2, -1)^T$
			$\mathbf{z}_{y_i,9} = (7, 1)^T$

A.2 Confusion Tables for Ordinal Classifier Cascades

The following appendix provides the confusion tables for the experiments on the artificial datasets $d_1, \ldots, d_{10}$. For each dataset, the results of ordinal classifier cascades trained for $|\mathcal{Y}|! = 24$ possible class orders are shown. Those class orders that allowed minimal class-wise sensitivities higher than 60 % are highlighted by a yellow halo. In each figure the results are organized according to the utilized base classifiers.

- Figure A.1: Confusion tables of dataset d_1: neighboring.
- Figure A.2: Confusion tables of dataset d_2: center-surround.
- Figure A.3: Confusion tables of dataset d_3: repetition (total).
- Figure A.4: Confusion tables of dataset d_4: repetition (partial).
- Figure A.5: Confusion tables of dataset d_5: wave (total).
- Figure A.6: Confusion tables of dataset d_6: wave (partial).
- Figure A.7: Confusion tables of dataset d_7: layered.
- Figure A.8: Confusion tables of dataset d_8: grid.
- Figure A.9: Confusion tables of dataset d_9: parallel.
- Figure A.10: Confusion tables of dataset d_{10}: mixed.

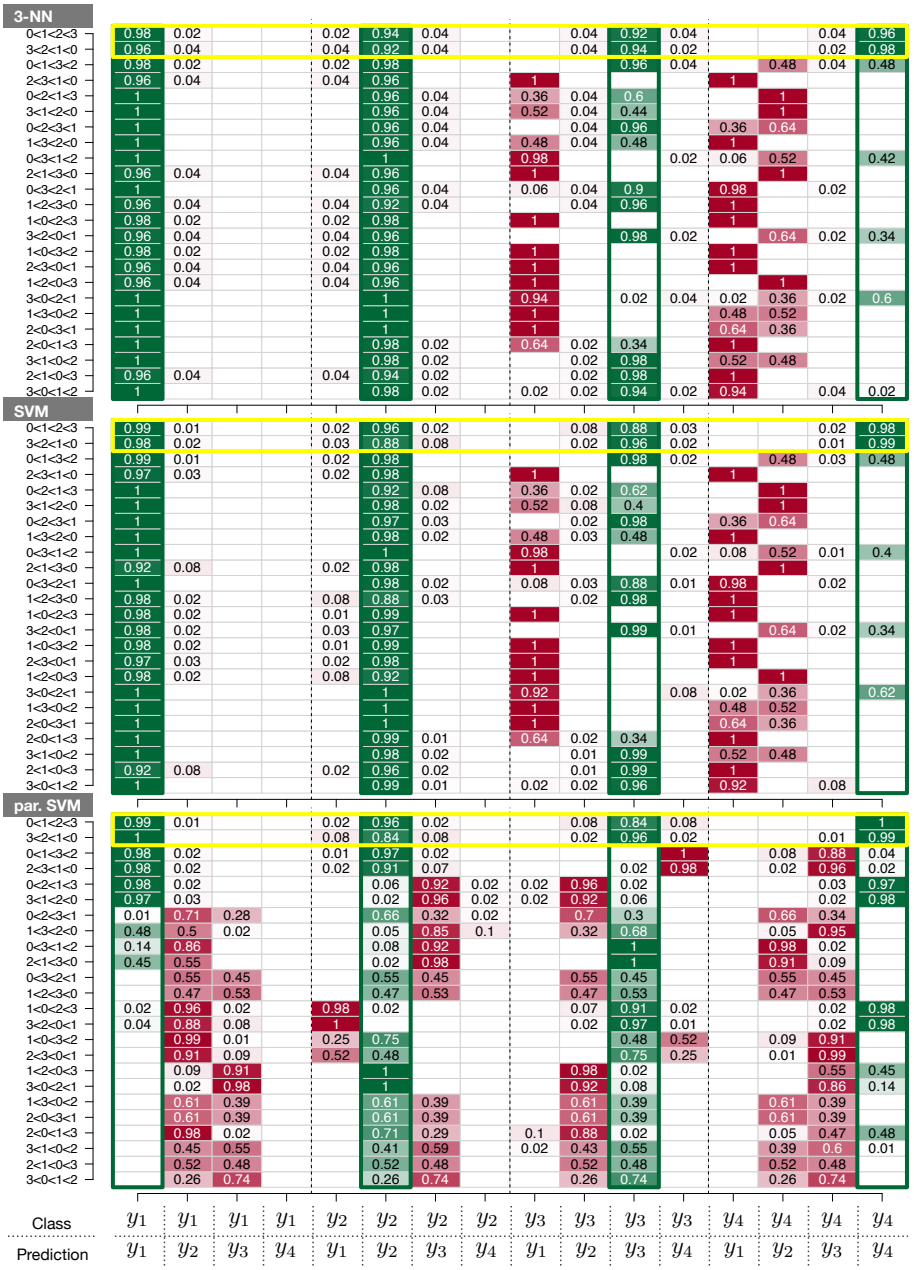


Figure A.1: Confusion tables of dataset d_1: neighboring.

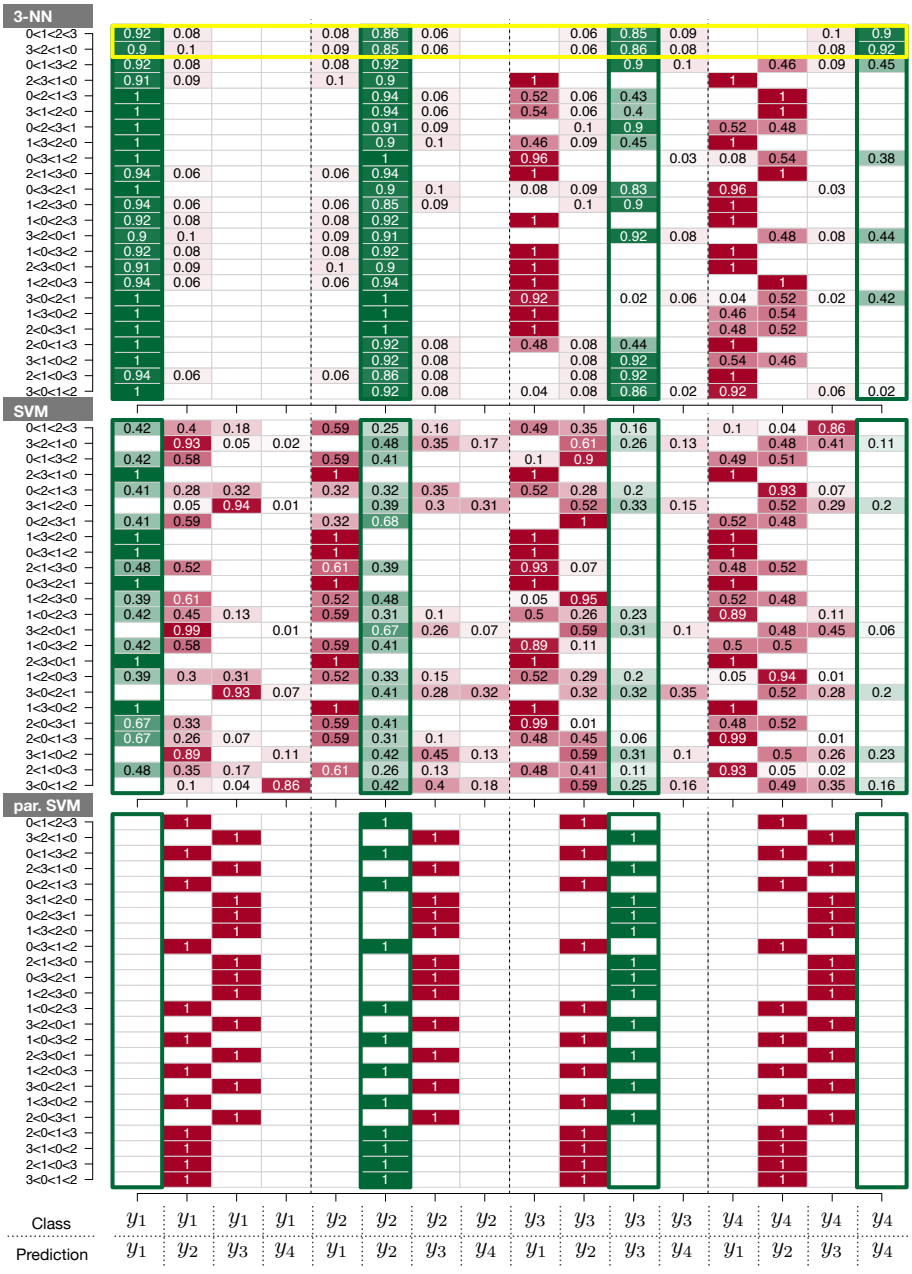


Figure A.2: Confusion tables of dataset d_2: center-surround.

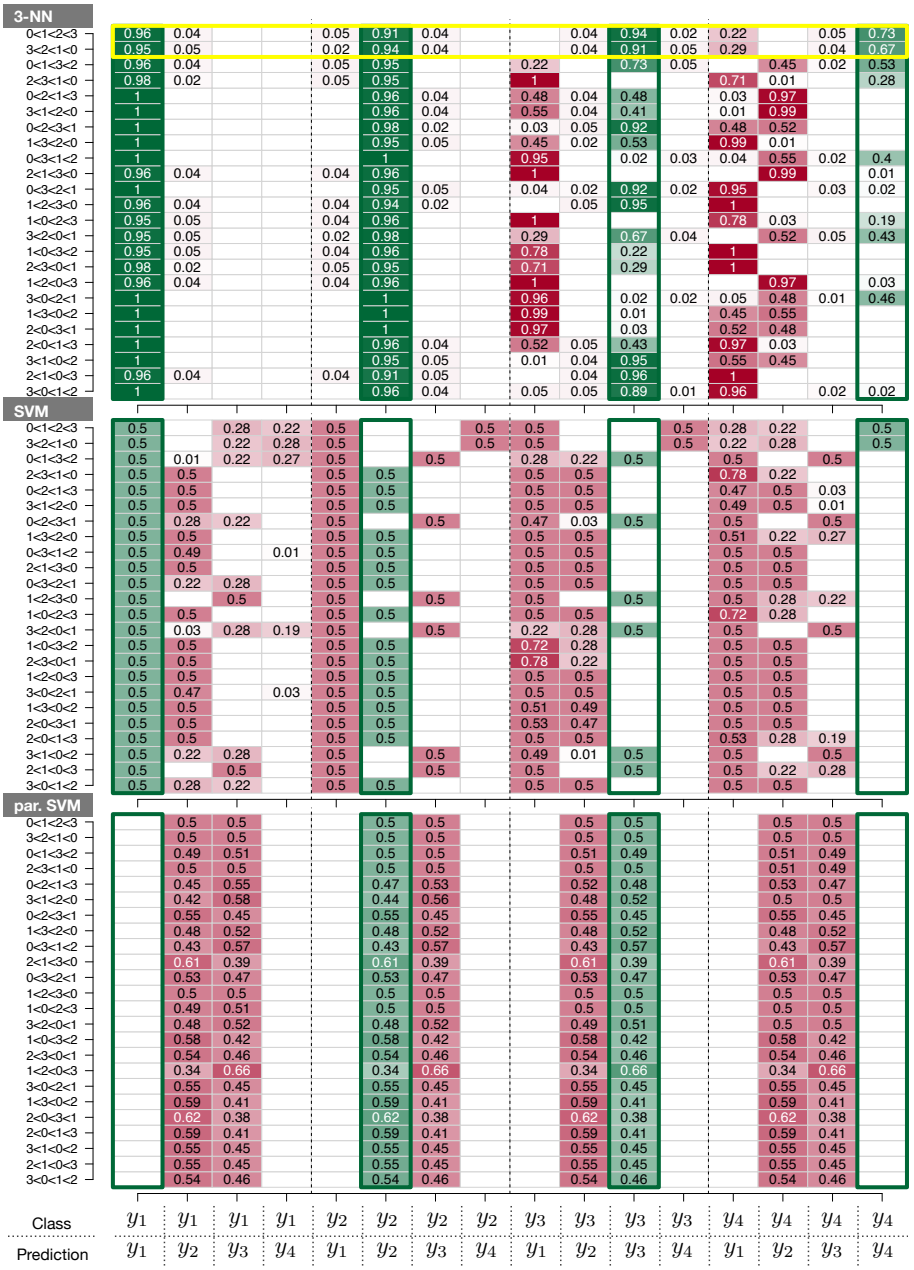


Figure A.3: Confusion tables of dataset d_3: repetition (total).

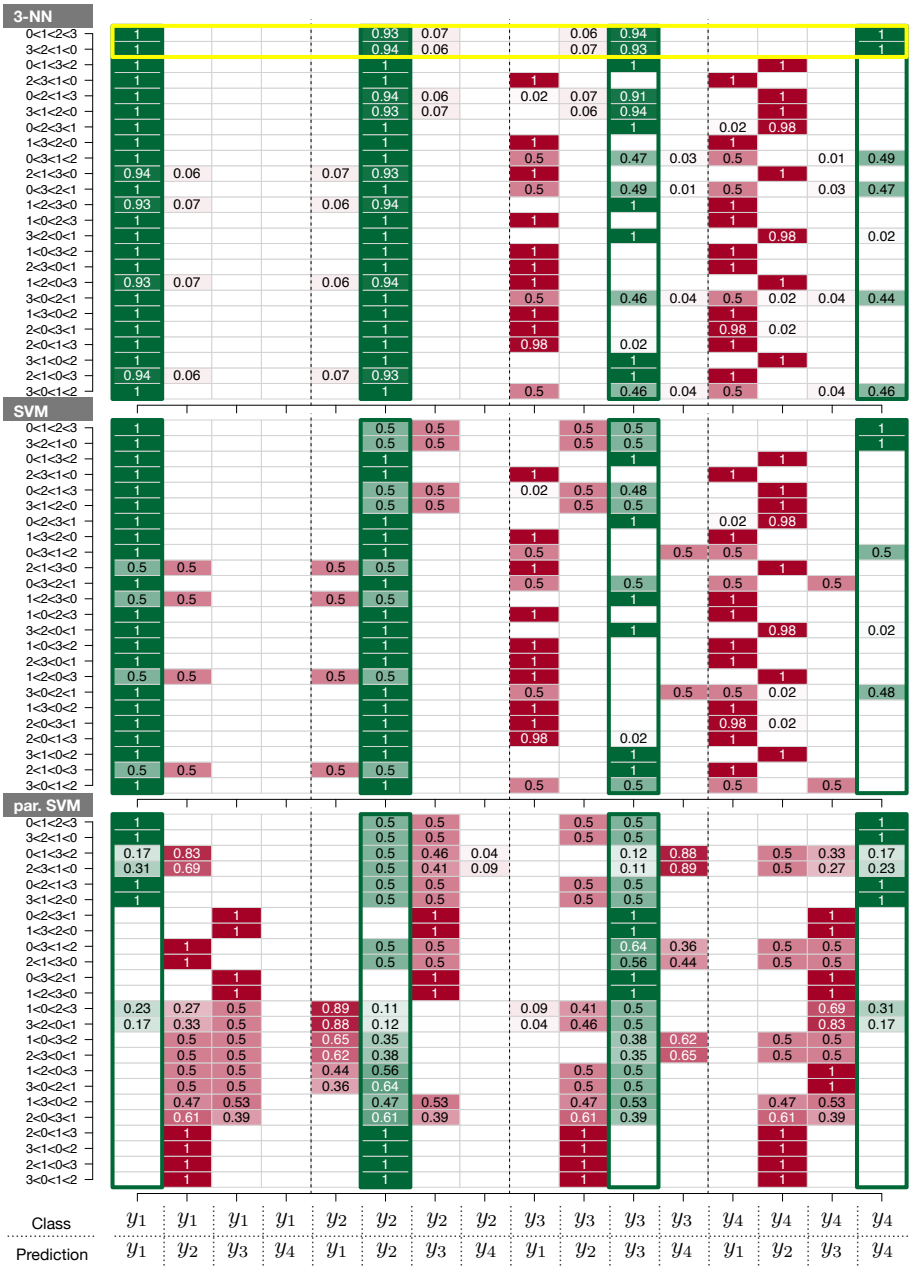


Figure A.4: Confusion tables of dataset d_4: repetition (partial).

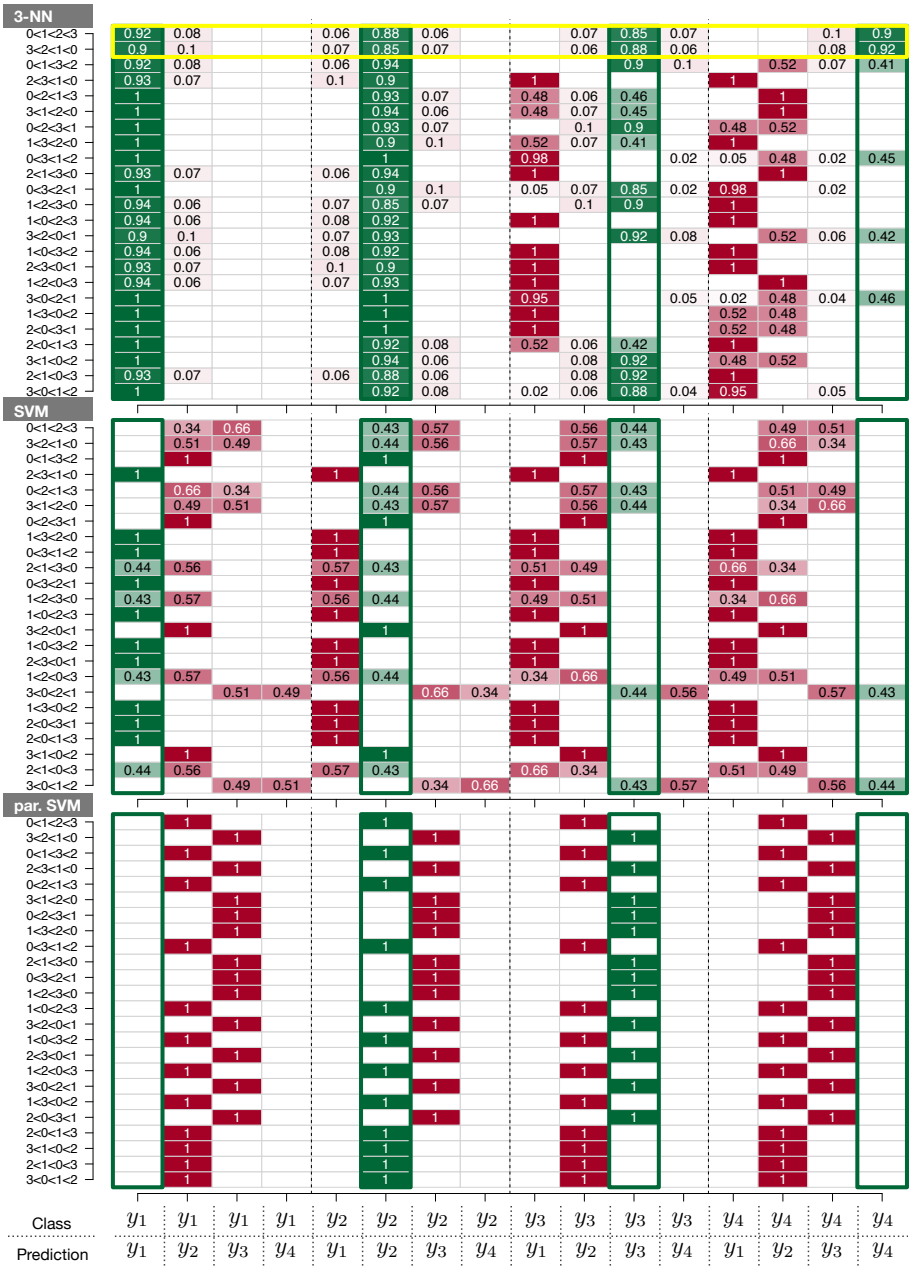


Figure A.5: Confusion tables of dataset d_5: wave (total).

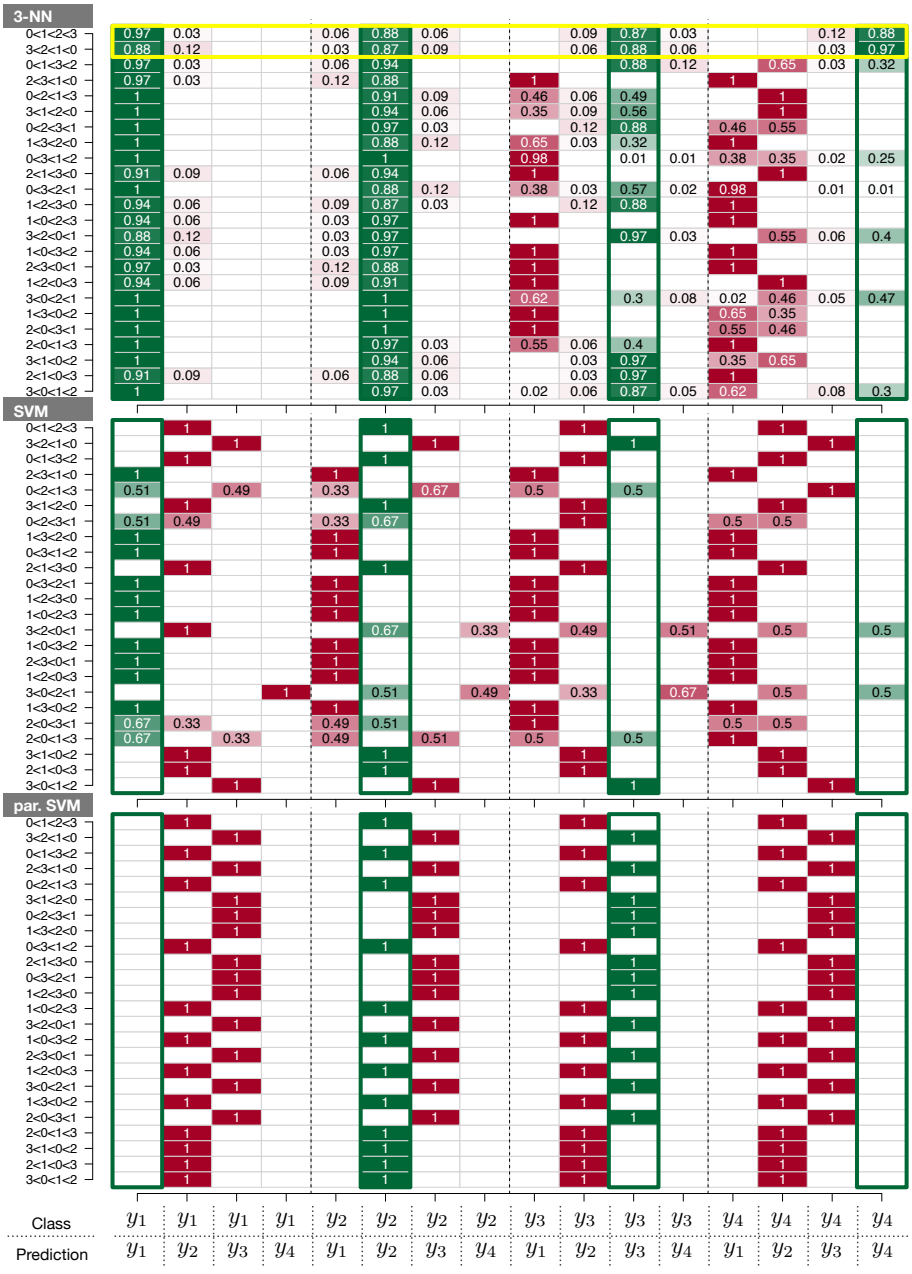


Figure A.6: Confusion tables of dataset d_6: wave (partial).

Figure A.7: Confusion tables of dataset d_7: layered.

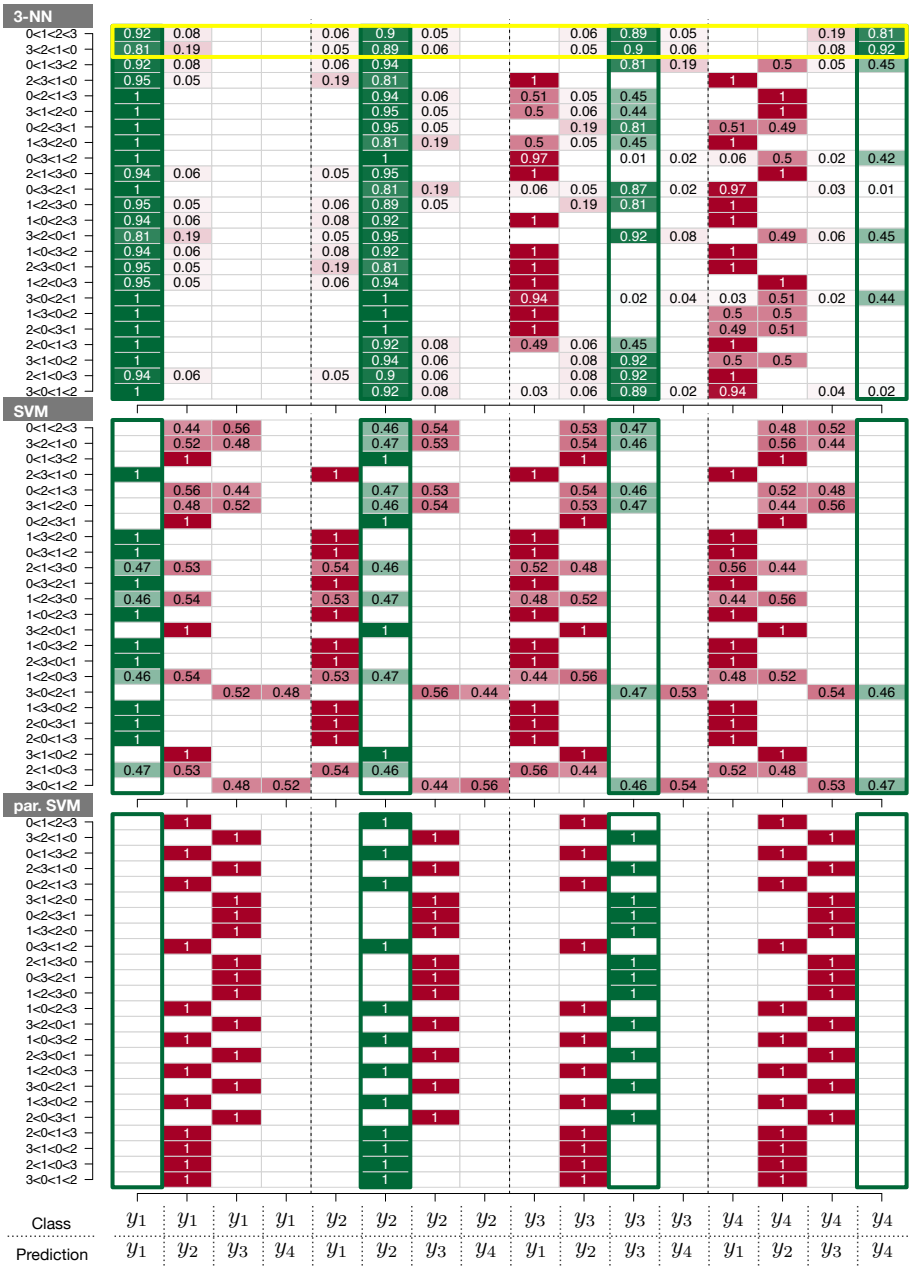


Figure A.8: Confusion tables of dataset d_8: grid.

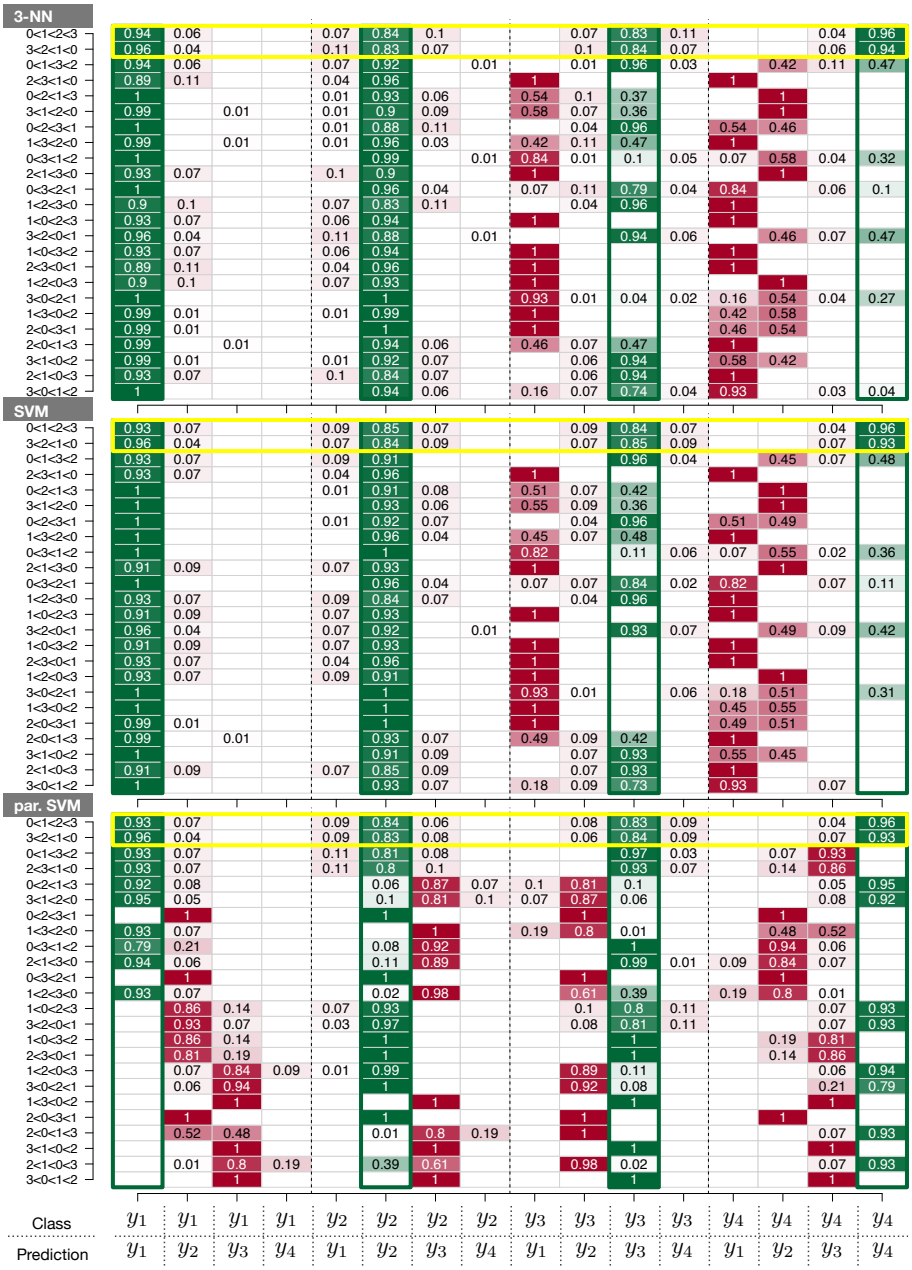


Figure A.9: Confusion tables of dataset d_9: parallel.

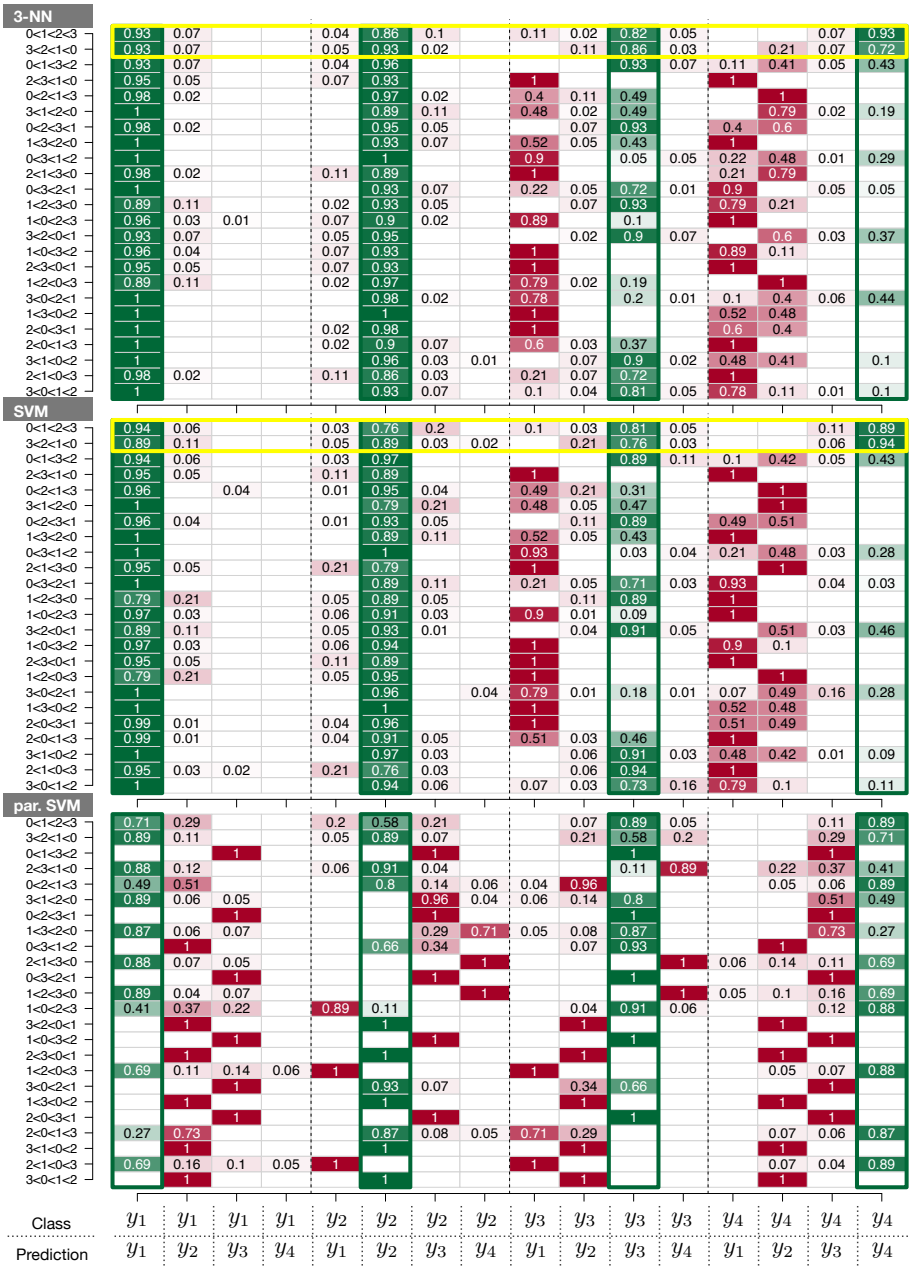


Figure A.10: Confusion tables of dataset d_{10}: mixed.