Ord2Seq: Regarding Ordinal Regression as Label Sequence Prediction

Jinhong Wang¹, Yi Cheng¹, Jintai Chen⁠†, TingTing Chen¹, Danny Chen², Jian Wu¹†
¹Zhejiang University  ²University of Notre Dame
{wangjinhong, chengyl, wujian2000}@zju.edu.cn
{jtchen721, ttchen0603}@gmail.com  dchen@nd.edu

Abstract

Ordinal regression refers to classifying object instances into ordinal categories. It has been widely studied in many scenarios, such as medical disease grading and movie rating. Known methods focused only on learning inter-class ordinal relationships, but still incur limitations in distinguishing adjacent categories thus far. In this paper, we propose a simple sequence prediction framework for ordinal regression called Ord2Seq, which, for the first time, transforms each ordinal category label into a special label sequence and thus regards an ordinal regression task as a sequence prediction process. In this way, we decompose an ordinal regression task into a series of recursive binary classification steps, so as to subtly distinguish adjacent categories. Comprehensive experiments show the effectiveness of distinguishing adjacent categories for performance improvement and our new approach exceeds state-of-the-art performances in four different scenarios. Codes are available at https://github.com/wjh892521292/Ord2Seq.

1. Introduction

Ordinal regression, a.k.a. ordinal classification, aims to classify object instances into ordinal categories. Since such categories follow a natural order, an ordinal regression task is typically treated as a classification problem with a few regression properties. Common applications are medical image grading [9, 23] (e.g., cataract can be graded from 0 to 6, representing normal to severe states), age estimation [29, 32, 21, 44], historical image dating [30, 28], and image aesthetic grading [15, 16, 31].

Unlike general classification tasks, it is challenging to distinguish the adjacent categories due to their confusing data patterns and blurred boundaries in ordinal regression tasks. Previous works often highlighted the ordering relations by introducing K-rank algorithms [12, 19, 29, 6], ordinal distribution constraint assumptions [22, 26, 17, 20], soft labels [13, 10], or multi-instance comparing approaches [24, 25, 20, 38]. However, these methods failed to specifically tackle the “adjacent categories distinction” and hinder the model performances.

In this paper, we argue and validate the importance of the “adjacent categories distinction” in ordinal regression tasks. To this end, we propose to distinguish the adjacent categories gradually in processing. Motivated by the dichotomic search (binary search) [45], which repeatedly divides the half portion of a sorted array to gradually find the target item, we decompose an ordinal regression task into a series of dichotomic classification steps. In each step, we can only focus on dealing with a boundary of a pair of adjacent categories. An example is given in Fig. 1. The aesthetics score of an image is gradually distinguished via recursive dichotomic classification. In this way, an ordi-

Figure 1. Our motivation. The dichotomic search (binary search) aims to repeatedly divide the half portion of a sorted array to find the target item. It can be utilized in ordinal regression tasks since the ordinal candidate labels can be regarded as a finite sorted array. Thus, an ordinal regression task is decomposed into multiple recursive dichotomic classification sub-problems. For example, when scoring an aesthetic image (e.g., from 1 to 5, the ground truth is 4), we can first estimate whether the score is above or below average (i.e., 3). Next, if it is above average, then we can further determine the score to be 4 or 5.
nal regression problem can be transformed into a sequence prediction problem that sequentially conducts dichotomic classification to finally obtain the ordinal category label.

Evolved from our motivation, we propose a simple sequence prediction framework for ordinal regression, called Ord2Seq. In our approach, ordinal regression is regarded as a sequence prediction task where the predicting goal is changed from a category label to a binary label sequence. That is, the prediction task is decomposed into a series of recursive binary classification steps to better distinguish adjacent categories in a process of progressive elaboration. Specifically, Ord2Seq performs two main steps. First, in pre-processing, we transform ordinal regression labels into label sequences by a tree-structured label mapping approach (we call the tree structure dichotomic tree in this paper). Thus, for each input data, the prediction objective turns to a sequence of binary labels. Next, it predicts this label sequence progressively via an encoder-decoder structured Transformer architecture. The Transformer is allowed to integrate context information by delivering the earlier image features and prediction results for the next token prediction. Also, the Transformer adapts to any sequence prediction length, so that our model has strong scalability on different tasks with various numbers of categories. Further, to enable our model to focus on each binary decision when distinguishing the remaining categories, the Transformer decoder is designed with a masked decision strategy to suppress the loss interference of the eliminated categories. Comprehensive experiments validate the superiority of our proposed Ord2Seq that carefully distinguishes adjacent categories.

Our main contributions are summarized as follows:

• For the first time, we propose to transform ordinal category labels as label sequences using a dichotomic tree, so as to tackle an ordinal regression task as a sequence prediction task.

• We propose a new sequence prediction framework for ordinal classification, called Ord2Seq, which effectively distinguishes adjacent categories with a process of progressive elaboration.

• We design a novel decoder with a masked decision strategy to suppress the loss interference of the eliminated categories in order to focus on distinguishing the remaining categories.

• Extensive experiments show the effectiveness of each component and that Ord2Seq performs better in distinguishing adjacent categories and achieves state-of-the-art performances on various image datasets.

2. Related Work

2.1. Ordinal Regression

The $K$-rank method [12] is the most popular approach for ordinal regression, in which $K - 1$ classifiers are trained to rank ordinal categories. A study [19] combined mathematical analysis based on the $K$-rank method to better learn inter-class ordinal relationships. Some methods [29, 6] used trained convolutional neural networks as $K$-rank classifiers. Although $K$-rank methods use a series of binary classifiers, their classifiers and classification process are independent, with no information transfer or interaction between these classifiers thus resulting in loss of inter-class information. Many recent studies [22, 26, 17, 20] proposed ordinal distribution constraints to exploit the ordinal nature of regression. To add prior order knowledge to loss calculation, several methods [13, 10] created soft labels artificially by changing the distances between categories. A few advanced methods [24, 25, 20, 38] sorted tuples that are formed by two [25] or three [24, 20, 38] instances with ordinal categories so the ranks of the test instances can be estimated from instances with known ranks. However, these methods only focused only on learning inter-class ordinal relationships and tend to be towards a misunderstanding that the latent features of adjacent categories should be as similar as possible. Consequently, these methods failed to highlight the boundaries between adjacent categories and perform not well in distinguishing adjacent categories, which hence hindered performance improvement. In this paper, we propose a dichotomy-based method to decompose ordinal regression into a series of recursive binary classification steps. Unlike the independent binary classifiers of $K$-rank methods. The binary classifier of our Ord2Seq has access to the predictions of the previous step and uses them to make further detailed predictions. With the candidate categories gradually refined, the model is able to focus on distinguishing adjacent categories.

2.2. Sequence Prediction

Sequence prediction was first applied in the natural-language processing field (e.g., machine translation [40, 1]). After Transformers [41] were shown to have powerful capabilities in sequence prediction, many Transformer models were developed for sequence prediction [33, 34, 3], and were also gradually introduced to computer vision (CV) [11]. But, in many CV tasks, Transformers were used only for feature extraction [27, 42]. Inspired by the success of transforming different domains into sequence prediction [5, 35], a few studies treated CV tasks as sequence prediction [7, 8], and showed considerable effectiveness. In these methods, Transformers were used to not only extract features but also predict sequences that are related to the target CV tasks. Our work is also inspired by previous sequence prediction models based on the Transformer architecture. With the sequence prediction scheme, we achieve to bring the idea of dichotomic search into the ordinal regression task for the first time by accessing the predictions of the previous step and making further detailed predictions.
3. Methodology

3.1. Overview

Our proposed Ord2Seq model takes an image $I$ as input, and transforms the ground truth (ordinal category labels) into binary label sequences in order to regard ordinal regression as a sequence prediction task. Thus, the prediction goal becomes to output a sequence of binary labels, as shown in Fig. 2. Ord2Seq consists of four main parts:

- **Label Transformation and Multi-hot Label Generation**: We construct a dichotomic tree for pre-processing, which transforms ground truth (ordinal category labels) into a sequence of binary label, and then generates a sequence of multi-hot labels for loss calculation.
- **Adaptive Encoder**: We utilize an Adaptive Encoder to extract imaging features, which is compatible with both CNN and Transformer backbones.
- **Masked Decision Decoder**: Our Masked Decision Decoder can directly predict probability sequences and indirectly predict binary label sequences with a masked decision strategy (one token at a time).
- **Loss Function**: Our model is trained to minimize the sum of the binary cross-entropy (BCE) losses of matched pairs between predicted probability sequences and generated multi-hot label sequences.

3.2. Label Transformation and Multi-hot Label Generation

**Label Transformation via a Dichotomic Tree.** Based on the dichotomy algorithm, we design a dichotomic tree to transform each ordinal category label into a sequence of binary label tokens for pre-processing. In this tree, the option paths to the left and right subtrees of each node are denoted by 0 and 1, respectively. If the number of categories is a power of 2, we construct a complete binary tree by dichotomy, as shown in Fig. 3(a). However, when the number of categories is not a power of 2, we cannot ensure that the numbers of categories in the two subtrees of every node are the same. Therefore, we construct an incomplete dichotomic tree in which the left and right subtrees of every node do not differ by more than one node and the depths of the leaf nodes for each category are equal, as shown in Fig. 3(c). After the tree construction, every category label $C$ is mapped to a corresponding binary label sequence $y_b$, showing an option path in the tree from the root node to the leaf node for the category $C$, by:

$$y_b = f(C) = [c_1, c_2, \ldots, c_d],$$

where $c_i \in \{0, 1\}$ denotes the codes of the option path for the category $C$, and $d$ donates the height of the tree.

Based on the constructed dichotomic tree, an ordinal category label is transformed into a binary label sequence, and our prediction target changes from a category label to a binary label sequence. Then, to predict the first label in sequence, following the shifted right process in vanilla Transformer [41], we shift the binary label sequence $y_b$ right with a starting query token $s$:

$$y_{\text{target}} = [s, c_1, c_2, \ldots, c_{d-1}].$$

**Multi-hot Label Generation.** Different from the language models, we do not directly predict the binary label sequence since such binary labels may hinder the model prediction for two reasons. (1) The 0’s and 1’s at different positions in our binary sequence may have different meanings. Thus, the model cannot forecast them directly. (2) The scope and meaning of each binary classification are different, and the classifiers should differentiate. For (1), we use

Figure 2. An overview of our Ord2Seq approach. Given an input image (e.g., for aesthetic grading), Ord2Seq transforms ordinal category labels into a binary label sequence so that the prediction target becomes a label sequence rather than an independent category label.
a Label Embedding approach (presented in Section 3.3) to map different 0’s and 1’s into different embeddings. For (2), based on the built dichotomic tree, we generate multi-hot label sequences which specify the scope and meaning of each classification. This process can be viewed as conducting continuous range predictions for the ground truth. Then the binary labels can be indirectly obtained from the range prediction results. Examples of Multi-hot Label Generation are shown in Fig. 3(b) and Fig. 3(d). Each node of the tree corresponds to a multi-hot label, and every category label $C_i$ is mapped to a corresponding multi-hot label sequence, as:

$$y_{mht} = g(C) = [a_0, a_1, \ldots, a_n], \quad (3)$$

where $a_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,n}]$ denotes the multi-hot labels of the path for the category $C_i$ with $a_{i,j} \in \{0, 1\}$ and $n$ being the number of categories. Thus, the multi-hot label at each node includes positive and negative classes, where the positive class is defined as the categories that the node includes and the negative class is for the other categories. With the supervision of multi-hot label sequences, the model can first predict a probability sequence and then output the binary label sequence based on the predicted probability sequence.

### 3.3. Masked Decision Decoder for Sequence Prediction

The masked decision decoder takes imaging features $X$ obtained by the Adaptive Encoder and a target sequence $y_{target}$ as input, predicts a probability sequence $y_{prob}$, and outputs a binary label sequence $y$. Fig. 4(a) overviews the masked decision decoder with its three main parts: Label Embedding, Transformer Decoder, and Masked Decision.

#### Label Embedding

To enable different 0’s and 1’s in each binary label sequence to represent different meanings, similar to the Position Encoding in [11], we use a function $h$ to map the target binary label sequence $y_{target}$ to a new vector with different values, and then encode the vector to the embeddings $y_{embd}$ with the same size of Transformer tokens via an embedding layer $E$, which can be formulated as:

$$h(y_{target}) = 2 \times i + y_{target}$$

$$y_{embd} = E(h(y_{target})). \quad (4)$$

Then $y_{embd}$ can be fed into Transformer Decoder as the latent target sequence.

#### Transformer Decoder

Our Transformer decoder $D$ follows the vanilla architecture [41] composed of Multi-headed Self-Attention (MSA), Layer Normalisation (LN), and Multi-headed Cross-Attention (MCA) layers with residual connections, aiming to predict the original logits sequence. For a time step $t$, the decoder $D$ takes the $t^{th}$ embedding token $y_{embd}^t$ as the input query $y_{in}^t$ and then sends it to the MSA and MCA layers, and a linear layer $w_t$, in sequence, to finally produce the original logits $y_{out}^t$, where MSA takes the previous input $y_{in}^{t-1}$ to compute keys and values, and MCA takes imaging features $X$ for attention calculation. We formulate the process at time step $t$ as:

$$y_{in}^t = \begin{cases} y_{embd}^t & \text{if training,} \\ E(h(y_{in}^{t-1})) & \text{if testing.} \end{cases}$$

$$y_{hidden,1}^t = \text{LN}(\text{MSA}(y_{in}^t W_Q; y_{in}^t W_K; y_{in}^t W_V)), \quad (5)$$

$$y_{hidden,2}^t = \text{LN}(\text{MCA}(y_{hidden,1}^t; X)),$$

$$y_{out}^t = y_{hidden,2}^t w_t^T,$$
where $W_Q$, $W_K$, and $W_V$ are weight matrices for computing queries, keys, and values. The logits $y_t^{\text{out}}$ are then used to generate a probability prediction $y_t^{\text{prob}}$ and a binary label sequence $y_t$ via the masked decision strategy (discussed below).

Note that during testing, the decoder $D$ takes the predicted binary label $y_{t-1}$ after Label Embedding as the input query $y_t^{\text{in}}$.

**Masked Decision.** The Masked Decision strategy is used to transform the original logits $y_t^{\text{out}}$ to a probability sequence $y_t^{\text{prob}}$ and predict a binary label sequence $y_t$ where the probability sequence $y_t^{\text{prob}}$ is for loss calculation and the binary label sequence $y_t$ is our prediction goal. By default, we perform the probability sequence prediction by $y_t^{\text{prob}} = \text{sigmoid}(y_t^{\text{out}})$. But obviously, for each time step $t$, the prediction should be based on the previous results. Thus, we try to suppress the loss interference of the eliminated categories in the previous prediction (time step $t-1$) with a mask. As shown in Fig. 4(b), for a time step $t$, the mask is defined as:

$$\text{Mask}_{t,i} = \begin{cases} 1 & y_{mht}^{t-1,i} = 1, \\ \alpha & y_{mht}^{t-1,i} = 0, \end{cases}$$  \hspace{1cm} (6)

where $\alpha$ is a hyper-parameter (we set $\alpha = 0.3$). Then the probability prediction at time step $t$ becomes:

$$y_t^{\text{prob}} = \text{Mask}_t \odot \text{sigmoid}(y_t^{\text{out}}),$$  \hspace{1cm} (7)

where $\odot$ is the element-wise product. Since $\alpha < 1$, the mask can be used to reduce the probability value of the $i^{th}$ category that satisfies $y_{mht}^{t-1,i} = 0$ (because all such categories have been eliminated in previous steps). Hence, the loss interference of these eliminated categories is restrained when calculating the loss between the predicted probability sequence $y_t^{\text{prob}}$ and the multi-hot sequence $y_{mht}$, forcing the model to focus on distinguishing the remaining categories.

After the masking process, we apply a decision strategy to predict the binary label based on the unmasked categories in $y_t^{\text{prob}}$ (see Fig. 4(b)). Suppose the categories of the left subtree are in $[l, m]$ and the categories of the right subtree are in $[m+1, r]$. We compute the average of all the probability values in each subtree, and compare them. According to the comparison result, we obtain the binary label $y_t$ for time step $t$. This process can be formulated as:

$$P_{\text{left}}^t = \frac{1}{m-l+1} \sum_{i=l}^{m} y_{mht}^{t,i},$$

$$P_{\text{right}}^t = \frac{1}{r-m} \sum_{i=m+1}^{r} y_{mht}^{t,i},$$

$$y_t = \begin{cases} 0 & P_{\text{left}}^t \geq P_{\text{right}}^t, \\ 1 & P_{\text{left}}^t < P_{\text{right}}^t, \end{cases}$$  \hspace{1cm} (8)

where $P_{\text{left}}^t$ and $P_{\text{right}}^t$ denote the average of the probability values of the categories in the left and right subtrees, respectively. The obtained binary label will be used for the next label prediction. As more binary labels are predicted, the remaining candidate categories are gradually dwindling and the adjacent categories are finally distinguished with higher confidence. After all the steps, we can inverse-map the resulted binary label sequence to the true category:

$$y_{\text{pred}} = f^{-1}(y).$$  \hspace{1cm} (9)

It can be seen that our masked decision decoder effectively joins the sequence prediction and decision-making process by first predicting a probability sequence via the Transformer decoder and then predicting a binary label sequence via the masked decision strategy.

**3.4. Other Details**

**Adaptive Encoder.** Our plug-and-play method adapts to any encoder-decoder architecture. Most existing vision Transformers are suitable as our encoder. In this work, we choose PVTv2 [43] as the encoder. Further, to adapt to popular CNN encoders such as VGG [39], we follow [4] by flattening the feature map after stage 5 (DC5); then the feature map is transformed to 512 channels and is passed to a Transformer encoder to obtain the imaging features $X$.

**Loss Functions.** Unlike the commonly used Cross Entropy (CE) loss in most ordinal classification methods, we choose Binary Cross Entropy (BCE) loss since our multi-hot labels have multiple positive classes which can be regarded as a multi-label classification problem for which CE loss is not suitable while BCE loss is. We first calculate the BCE loss between $y_t^{\text{prob}}$ and $y_{mht}^{t}$ at each time step $t$, and then sum them up as:

$$L = \sum_{t=1}^{d} BCE(y_t^{\text{prob}} | y_{mht}^{t}) = -\frac{1}{n} \sum_{t=1}^{d} \sum_{i=1}^{n} (y_{mht}^{t,i} \log(y_{t}^{i}) + (1 - y_{mht}^{t,i}) \log(1 - y_{t}^{i})).$$  \hspace{1cm} (10)
Figure 5. Ord2Seq (PVT) performances with different values of the mask α on the Adience dataset. It achieves the best performance when α = 0.3.

Table 1. Ablation experiments on the Adience dataset. For Accuracy, higher is better; for MAE, lower is better. †denotes the Ord2Seq model without the masked decision strategy.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG only</td>
<td>57.4</td>
<td>0.55</td>
</tr>
<tr>
<td>VGG + Trans</td>
<td>57.8 (+0.4)</td>
<td>0.51 (-0.04)</td>
</tr>
<tr>
<td>Ord2Seq (VGG)†</td>
<td>61.6 (+4.2)</td>
<td>0.49 (-0.06)</td>
</tr>
<tr>
<td>Ord2Seq (VGG)</td>
<td><strong>63.5</strong> (+6.1)</td>
<td><strong>0.44</strong> (-0.11)</td>
</tr>
</tbody>
</table>

Table 2. Accuracy and MAE comparison on the Adience dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>MAE</th>
<th>Inf time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWR [38]</td>
<td>62.6</td>
<td>0.45</td>
<td>1803</td>
</tr>
<tr>
<td>VGG+Trans</td>
<td>57.8</td>
<td>0.51</td>
<td>203</td>
</tr>
<tr>
<td>Ord2Seq (VGG)†</td>
<td>63.5</td>
<td>0.44</td>
<td>318</td>
</tr>
</tbody>
</table>

Table 3. Performance and inference time comparison on the Adience dataset. For inference time, lower is better.

4. Experiments

To validate the effectiveness of our Ord2Seq approach, we conduct extensive experiments on the datasets of four different scenarios: Image Aesthetics, Age Estimation, Historical Image Dating, and Diabetic Retinopathy Grading.

4.1. Experimental Setup

Our experiments use a computer with an Intel i7 processor and an NVIDIA GTX 2080Ti GPU. To compare with existing methods that use VGG-16 as the backbone, we train Ord2Seq with two Adaptive Encoders, VGG-16 [39] and PVTv2-b1 [43], with similar settings and pre-trained on ImageNet [37]. The mini-batch size is 32. We use random horizontal flipping and random cropping to the crop size of 224 × 224 for data augmentation. For optimization, the Adam optimizer [14] is utilized with a learning rate of 10^{-4}. For the Mean Average Error (MAE) metric, it is computed by the expectation value of the predictions and the target value. For fair comparisons, all the known methods are implemented using the authors’ code or re-implemented based on the original papers. More details about the datasets and experimental settings are in the supplemental document.

4.2. Age Estimation

Dataset: The Adience dataset [18] is used for age group estimation that contains about 26,580 face images from Flickr of 2,284 subjects. Ages are annotated in 8 groups: 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, and over 60 years old. All the images are divided into 5 subject-exclusive folds for cross-validation as in [25, 10, 20, 38].

**Comparison with Known Methods:** We show the comparison results on the Adience dataset in Table 2. We find
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%) – higher is better</th>
<th>MAE – lower is better</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nature</td>
<td>Animal</td>
</tr>
<tr>
<td>CNNPOR [25]</td>
<td>71.86</td>
<td>69.32</td>
</tr>
<tr>
<td>SORD [10]</td>
<td>73.59</td>
<td>70.29</td>
</tr>
<tr>
<td>POE [20]</td>
<td>73.62</td>
<td>71.14</td>
</tr>
<tr>
<td>Ours (VGG)</td>
<td>78.22</td>
<td></td>
</tr>
<tr>
<td>Ours (PVT)</td>
<td>78.09</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Results on the Image Aesthetics dataset. Accuracy and MAE are reported for each of the four image classes.

that our Ord2Seq with the VGG encoder achieves better results than the existing methods that use the same VGG architecture. Compared with POE [20] and SORD [10], in despite slightly larger size, Ord2Seq achieves significant performance gains. Compared with the previous SOTA model MWR [38], our Ord2Seq achieves a superior performance while the model sizes are much smaller than MWR. These comparison results demonstrate the superiority of our proposed approach by focusing on distinguishing adjacent categories. Further, Ord2Seq with the PVT encoder achieves an accuracy of 63.9% and MAE of 0.43, defeating all current SOTA results, which shows that a sequence-to-sequence skeleton (using a Transformer encoder as the backbone) can exert the potential of our approach.

**Inference Time Analysis:** We compute the average inference time (ms) of our methods and MWR for predicting 1 batch image on the Adience dataset. The results are shown in Table 3. As well as in Table 1, VGG+Trans denotes the model that keeps the Transformer decoder structure but removes the sequence prediction scheme to only predict a true label once. Ord2Seq (VGG) denotes our sequence prediction framework. It can be observed that (1) Although our Ord2Seq requires multi-step Transformer decoder inference, its inference time increases only by about half of the inference time of VGG+Trans, which shows that most of the time the model takes is in feature extraction instead of decoder inference. (2) The inference time of our Ord2Seq is significantly less than MWR but performance is improved, which further validates the superiority of Ord2Seq.

### 4.3. Image Aesthetics

**Dataset:** The Aesthetics dataset [11] contains 15,687 Flickr image URLs, 13,706 of which are available. The dataset is used to grade image aesthetics. There are four image classes: animals, urban, people, and nature. Each image was graded by at least 5 different graders in 5 ranking categories: unacceptable, flawed, ordinary, professional, and exceptional. The ground truth is defined as the median rank among all the gradings. Following [25, 10, 20], we apply 5-fold cross-validation. The images are randomly divided by 75%, 5%, and 20% for training, validation, and testing, respectively.

**Results:** Table 4 shows the results on the Image Aesthetics dataset. We observe that our approach significantly outperforms the existing methods in various metrics. For example, our model with the PVT encoder achieves an Accuracy of 78.09% for the Nature class, an overall Accuracy of 74.43%, and an overall MAE of 0.264, outperforming the POE method [20] by 4.47% for the Nature class, 1.99% for the overall Accuracy, and 0.023 for the overall MAE. Except the mediocre performances of our model for the People class that may be due to that people’s aesthetics is more subjective with various factors (e.g., gender, age, expression,
Table 6. Accuracy and MAE comparison on the DR dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson [2]</td>
<td>77.1 ± 0.6</td>
<td>0.38 ± 0.25</td>
</tr>
<tr>
<td>MT [36]</td>
<td>82.8 ± 0.6</td>
<td>0.36 ± 0.22</td>
</tr>
<tr>
<td>SORD [10]</td>
<td>78.2 ± 0.6</td>
<td>0.73 ± 0.17</td>
</tr>
<tr>
<td>POE [20]</td>
<td>80.5 ± 0.6</td>
<td>0.30 ± 0.21</td>
</tr>
<tr>
<td>Ours (VGG)</td>
<td>84.0 ± 0.6</td>
<td>0.25 ± 0.07</td>
</tr>
<tr>
<td>Ours (PVT)</td>
<td>84.2 ± 0.5</td>
<td>0.25 ± 0.07</td>
</tr>
</tbody>
</table>

Ord2Seq achieves state-of-the-art results for the other classes, which validate the effectiveness of our approach.

4.4. Historical Image Dating

Dataset: The historical color image (HCI) dataset is for estimating the decades of historical color photos. There are five decades from 1930s to 1970s annotated as 1 to 5. Each decade has 265 images. Following [25, 26, 20, 38], we randomly split the 265 images of each decade into three subsets: 210 for training, 5 for validation, and 50 for testing. Then 10-fold cross-validation is performed, and the mean values of the results are recorded.

Results: Table 5 compares the results on the HCI dataset. As can be seen, our Ord2Seq with the VGG encoder outperforms known methods that use the same VGG architecture. Further, Ord2Seq with the PVT encoder achieves state-of-the-art results, providing improvements of 3.1% in Accuracy and 0.06 in MAE, which indicate the superiority of our approach. In addition, to provide more details of the model performances in distinguishing adjacent categories, for all samples whose ground truths are of the same category, we calculate the proportions of these samples that are predicted to the correct, adjacent, and other categories. As visualized in Fig. 6, we find that, for most categories, although the sums of the correct and adjacent proportions attained by different methods are close, our proposed Ord2Seq achieves higher proportions of the correct predictions and lower proportions of the adjacent predictions. That is, Ord2Seq is able to successfully predict part of samples that tended to be predicted into adjacent categories by previous methods. This result shows the effectiveness of Ord2Seq in distinguishing adjacent categories, which is also the main performance improvement of our method comes from.

4.5. Diabetic Retinopathy Grading

Dataset: The Diabetic Retinopathy (DR) dataset contains 35,126 high-resolution fundus images available at https://www.kaggle.com/c/diabetic-retinopathy-detection. In this dataset, images were annotated in five levels of diabetic retinopathy from 1 to 5, representing no DR (25,810 images), mild DR (2,443 images), moderate DR (5,292 images), severe DR (873 images), and proliferative DR (708 images), respectively. Some sample images are shown in Fig. 7. Following the setting used in [2, 23], we apply the subject-independent 10-fold cross-validation, and report the mean values of the results.

Results: Table 6 shows the results on the DR dataset. Note that the DR dataset is unbalanced since the sample number decreases sharply as the severity DR level increases. We observe that the known order learning methods yield poor performances which may be due to the unbalanced data. Especially, SORD [10], which is a modality-specific method by utilizing modified soft labels, can suffer serious errors in MAE. In comparison, our proposed Ord2Seq still maintains competitive performances, achiev-
ing an Accuracy of 84.2% and an MAE of 0.25, which greatly outperforms the baselines and the other order learning methods, showing that our approach has better robustness on unbalanced data. We believe that this is due to the better positive-negative distinction. That is, unlike one positive class against other negative classes in previous work, it turns to (e.g.) classifier the first two categories against last three categories in the first step of Ord2Seq for the DR dataset (5 categories in total). In this way, the classification in a step is more category-balanced and helps to better exert unbalanced data. Moreover, to validate the superiority of Ord2Seq in distinguishing adjacent categories, we visualize the model performances based on the proportions of samples that truly belong to one category and are predicted to the correct, adjacent, and other categories on the DR dataset, in Fig. 8. It is obvious that for levels 2–5 (with limited numbers of samples), although all the compared methods yield sub-optimal performances, our Ord2Seq significantly improves the correct prediction proportions and reduces the adjacent prediction proportions. This result also validates that our approach has better generalization on unbalanced categories, and can effectively distinguish adjacent categories to achieve higher overall performance.

5. Conclusions

In this paper, we proposed a new sequence prediction framework for ordinal regression, Ord2Seq, which transforms ordinal labels as binary label sequences and uses a dichotomy-based sequence prediction procedure to distinguish adjacent categories based on a progressive elaboration scheme. Extensive experiments showed that Ord2Seq achieves state-of-the-art performances in various applied scenarios, and verified that Ord2Seq can effectively distinguish adjacent categories for performance improvement.

The insight of our approach, i.e. dichotomy-based sequence prediction, is instructive for other general classification tasks. By dividing similar categories into a subtree and gradually refining the classification through a sequence prediction process, our model may be able to effectively distinguish similar objects with fine-grained differences (e.g., donkeys and horses).

Acknowledgements. This research was partially supported by National Key R&D Program of China under grant No. 2018AAA0102102, National Natural Science Foundation of China under grants No. 62176231, No. 82202984, No. 62106218, No. 92259202, No. 62132017 and U22B2034, Zhejiang Key R&D Program of China under grant No. 2023C03053.

References


[42] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kai Tao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision Transformer: A versatile backbone for dense

