UNIVERSALLY RANK CONSISTENT ORDINAL REGRESSION IN NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the pervasiveness of ordinal labels in supervised learning, it remains common practice in deep learning to treat such problems as categorical classification using the categorical cross entropy loss. Recent methods attempting to address this issue while respecting the ordinal structure of the labels have resorted to converting ordinal regression into a series of extended binary classification subtasks. However, the adoption of such methods remains inconsistent due to theoretical and practical limitations. Here we address these limitations by demonstrating that the subtask probabilities form a Markov chain. We show how to straightforwardly modify neural network architectures to exploit this fact and thereby constrain predictions to be universally rank consistent. We furthermore prove that all rank consistent solutions can be represented within this formulation. Using diverse benchmarks and the real-world application of a specialized recurrent neural network for COVID-19 prognosis, we demonstrate the practical superiority of this method versus the current state-of-the-art. The method is open sourced as user-friendly PyTorch and TensorFlow packages.

1 INTRODUCTION

Ordinal regression (sometimes called ordinal classification) is applied to data in which the features of the n-th example $\boldsymbol{x}_n \in \mathbb{X}$ correspond to a label $y_n \in \mathbb{Y} := \{r_1, \ldots, r_K\}$ from a set of elements that have a well-defined ranking or ordering $r_1 < r_2 < \cdots < r_K$. However, unlike traditional metric regression, the ranks cannot be assumed to have quantitative differences or distances amongst themselves. For example, while a syntactic statement such as "terrible"<"great"<"best" may be intuitive, it conveys nothing about quantitative distance between the ranks nor qualitative differences i.e. whether the distance between "terrible" and "great' is equal to that between "great" and "best". The aim in this setting is to build a reliable rule or regression function $h : \mathbb{X} \to \mathbb{Y}$ from the domain of the features $\mathbb{X}$ to the range of the ordinal labels $\mathbb{Y}$.

In the published literature for applied problems, it remains commonplace to ignore the ordering of the labels and apply categorical algorithms to such data (Levi & Hassncer, 2015; Rothe et al., 2015), which in neural networks often results in application of the categorical cross entropy (CCE) loss. Problematically, a categorical loss assumes all mislabeling by h is equally wrong, whereas it is clear that predicting "great" when the true label is "best" would be preferable to a prediction of "terrible". Although the problematic nature of this practice has been recognized for more than 35 years (Forrest & Andersen, 1986), it still remains common to make the implicit or explicit assumption that ordinal data or labels exist on an interval or ratio scale.

Most contemporary algorithms found in the ordinal regression literature (McCullagh, 1980; Obermayer, 1999; Crammer & Singer, 2002; Shashua & Levin, 2002; Rajaram et al., 2003; Shen & Joshi, 2005; Chu & Keerthi, 2005; Li & Lin, 2007; Baccianella et al., 2009; Niu et al., 2016; Cao et al., 2020), can be viewed through the lens of a general framework proposed by Li & Lin (2007) wherein the labels are encoded as binary vectors by an invertible encoder $e : \mathbb{Y} \to \{0,1\}^{K-1}$ and the regression function h becomes a collection of $K-1$ binary classifiers along with the decoder $e^{-1} : \{0,1\}^{K-1} \to \mathbb{Y}$. However, many of these existing algorithms share one major limitation: rank-inconsistency among predictions. Briefly, the $K-1$ binary tasks are not independent and without special care, training $K-1$ classifiers will produce conflicting predictions on the binary tasks, impeding both performance and interpretation of the results. Most recently, Cao et al. (2020)

attempted to address this problem in deep neural network (DNN) architectures by proposing a final layer that shares weights among $K-1$ binary outputs (differing only in their bias terms).

Herein, we identify the theoretical and practical limitations of the CORAL approach by Cao et al. (2020) , and address these concerns by implementing a new algorithm 'Conditionals for Ordinal Regression' (CONDOR). We prove that CONDOR is universally rank consistent and that it is sufficiently expressive to reach any rank consistent solution. In theory, the method is compatible with any combination of binary classification algorithms, but herein we focus on DNN architectures trained by backpropagation. Using our open source PyTorch and TensorFlow packages, CONDOR can be implemented with minor modifications to any categorical DNN model in these frameworks, allowing for increased adoption of ordinal regression in the applied literature.

2 METHODS

We begin by introducing the CONDOR framework and its notations, and then provide proofs of the universal rank consistency and the full expressiveness of the framework.

2.1 CONDOR FRAMEWORK

The Li & Lin (2007) encoder e converts the ordinal regression label y into $K-1$ binary classification labels $y^{(1)}, \ldots, y^{(K-1)}$ using indicator variables $y^{(1)} := \mathbf{1}_{\mathrm{y}>\mathrm{r}_1}, \ldots, y^{(K-1)} := \mathbf{1}_{\mathrm{y}>\mathrm{r}_{\mathrm{K}-1}}$, where the indicator variable $\mathbf{1}_{\mathrm{expr}}$ is defined as

$$\mathbf{1}_{\mathrm{expr}} := \begin{cases} 1, & \text{expr is true,} \\ 0, & \text{expr is false.} \end{cases}$$

The ordinal classification problem then becomes a matter of producing $K-1$ binary classifier subtasks $f_k : \mathbb{X} \to \{0,1\}$, which we assume come from thresholding predicted binary class probabilities $p_k : \mathbb{X} \to [0,1]$ as $f_k(\boldsymbol{x}) = \mathbf{1}_{\mathrm{p_k(x)}>0.5}$.

For convenience, we often deal with the rank index $s \in \{1, \ldots, K-1\}$ associated with the rank $r_s \in \mathbb{Y}$. From the binary classifier subtasks, the rank index s_i for input feature vector $\boldsymbol{x}_i$ can be estimated as

$$s_i = 1 + \sum_{k=1}^{K-1} f_k(\boldsymbol{x}_i), \tag{1}$$

although multiple methods are possible to produce point estimates for s_i from the probabilities $p_k(\boldsymbol{x}_i), k = 1, \ldots, K-1$. As shown in Figure 1, the aforementioned binary encoding approach requires that $p_k, k = 1, \ldots, K-1$ be rank-monotonic

$$p_1(\boldsymbol{x}_i) \geq \cdots \geq p_{K-1}(\boldsymbol{x}_i),$$

for all $\boldsymbol{x}_i \in \mathbb{X}$ to guarantee consistent predictions.

Rather than directly estimating the marginal probabilities $p_k(\boldsymbol{x}) = P(\mathrm{y}^{(k)} = 1 | \mathbf{x} = \boldsymbol{x})$, $k = 1, \ldots, K-1$, as in existing approaches based on Li & Lin (2007), we estimate the conditionals

$$\begin{aligned} q_k(\boldsymbol{x}) &:= P(\mathrm{y}^{(k)} = 1 | \mathbf{x} = \boldsymbol{x}, \mathrm{y}^{(k-1)} = 1, \mathrm{y}^{(k-2)} = 1, \ldots, \mathrm{y}^{(0)} = 1) \\ &= P(\mathrm{y}^{(k)} = 1 | \mathbf{x} = \boldsymbol{x}, \mathrm{y}^{(k-1)} = 1), \end{aligned} \tag{2}$$

for $k = 1, \ldots, K-1$ where we set the boundary condition $\mathrm{y}^{(0)} = 1$ with unit probability by convention. Equality (2) follows by construction of the binary labels since $y^{(k-1)} = 1$ implies $y > r_{k-1} > r_{k-2} > \cdots > r_1$, which by definition means $y^{(k')} = 1$ for $k' \leq k-1$. By the same reasoning, the marginal probability is equivalent to the joint probability

$$p_k(\boldsymbol{x}) := P(\mathrm{y}^{(k)} = 1 | \mathbf{x} = \boldsymbol{x}) = P(\mathrm{y}^{(k)} = 1, \mathrm{y}^{(k-1)} = 1 \ldots, \mathrm{y}^{(0)} = 1 | \mathbf{x} = \boldsymbol{x})$$

and by the product rule we produce a heterogeneous Markov chain representation of our marginal probabilities

$$p_k(\boldsymbol{x}) = \prod_{k'=1}^{k} q_{k'}(\boldsymbol{x}). \tag{3}$$

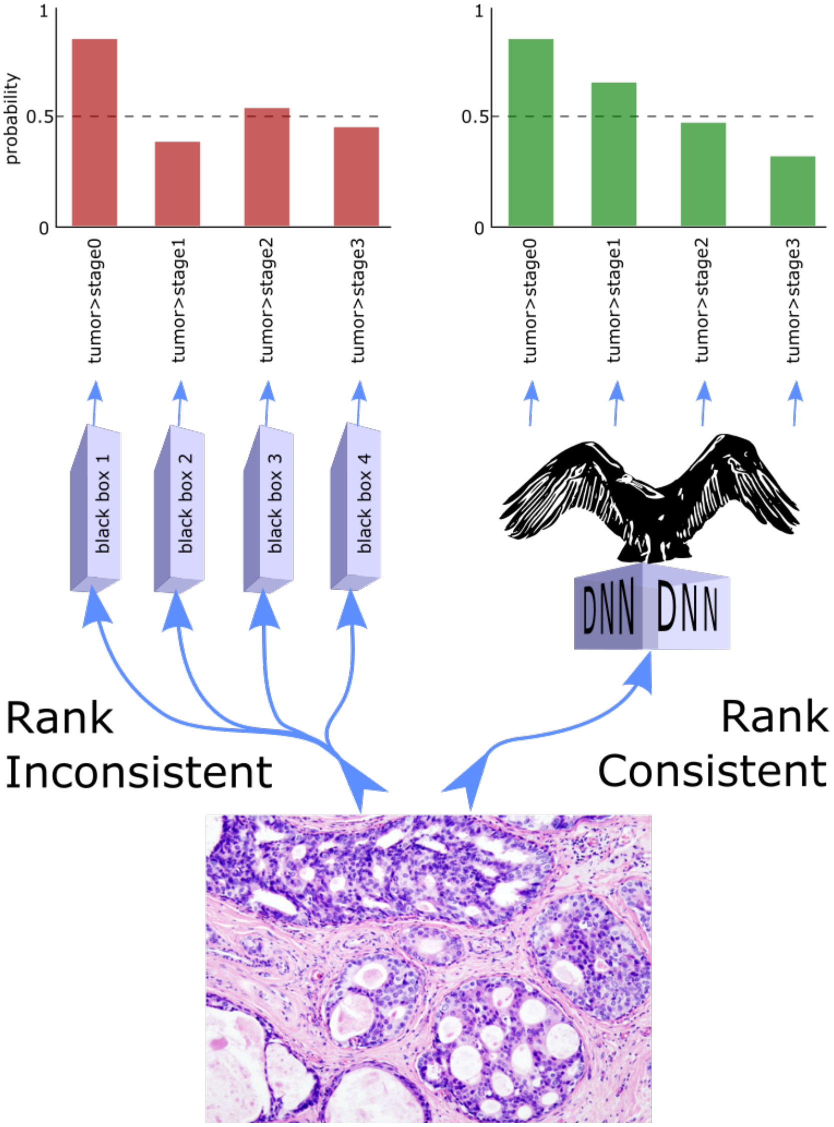


Figure 1: All existing methods can produce rank inconsistent predictions, whereas CONDOR sits atop any DNN architecture and produces universally rank consistent results. This improves performance and interpretability of the ordinal model.

The above method in Equations (2) & (3) are exact (i.e., not approximations) and thus fully general. They can in theory be applied in any classifier estimating binary probabilities, but we focus on the application to DNNs. Namely, we select the final layer of the neural network to have $K-1$ nodes with sigmoid activations representing $q_k(\boldsymbol{x}), k = 1, \ldots, K-1$. For training, the loss is the weighted binary cross-entropies of all the subtasks

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K-1}\lambda_k\left(y_n^{(k)}\ln\left[\prod_{k'=1}^{k}q_{k'}(\boldsymbol{x}_n)\right] + (1-y_n^{(k)})\ln\left[1-\prod_{k'=1}^{k}q_{k'}(\boldsymbol{x}_n)\right]\right) \tag{4}$$

where $\lambda_k > 0$ is the importance parameter for task k, which we set to one in the subsequent. We call this approach Conditionals for Ordinal Regression (CONDOR).

2.2 RANK CONSISTENCY

Here we substantiate CONDOR's guarantee for preserving rank consistency and its ability to represent any rank consistent solution.

Lemma 2.1. *CONDOR provides universal rank consistency (i.e., rank consistent estimates for all input data $\boldsymbol{x} \in \mathbb{X}$ and any parameterization of the DNN).*

Proof. In neural networks, we can enforce $0 < q_k(\boldsymbol{x}) < 1$ for all $\boldsymbol{x}$ for any weight parameterization of the DNN by having $K-1$ output nodes with sigmoid activations representing $q_k(\boldsymbol{x}), k = 1, \ldots, K-1$. Because $0 < q_k(\boldsymbol{x}) < 1$ for all $\boldsymbol{x}$, we have by Equation (3)

$$p_{k+1}(\boldsymbol{x}) = p_k(\boldsymbol{x}) q_{k+1}(\boldsymbol{x}) < p_k(\boldsymbol{x})$$

for all $\boldsymbol{x}$ and $k = 1, \ldots, K-2$. Thus we have rank consistency

$$p_1(\boldsymbol{x}) \geq p_2(\boldsymbol{x}) \geq \cdots \geq p_{K-1}(\boldsymbol{x})$$

for all $\boldsymbol{x}$ and any weight parameterization of the DNN. □

Theorem 2.2. *Assuming that a neural network can universally approximate any $\mathcal{C}^1$ function $g : \mathbb{X} \to \mathbb{R}^{K-1}$ such that $\widehat{g}(\boldsymbol{x}) = g(\boldsymbol{x}) + \mathcal{O}(\epsilon)$ for all $\boldsymbol{x}$ and some $\epsilon > 0$, then adding a CONDOR output layer to said network can approximate any rank consistent continuous ordinal regressors $p_k^* : \mathbb{X} \to [0,1], k = 1, \ldots, K-1$ with error $\mathcal{O}(\epsilon)$.*

Proof. By rank consistency

$$p_1^*(\boldsymbol{x}_i) \geq \cdots p_{k-1}^*(\boldsymbol{x}) \geq p_k^*(\boldsymbol{x}) \cdots \geq p_{K-1}^*(\boldsymbol{x}_i) \geq 0$$

for any $\boldsymbol{x}$ and we have defined the boundary condition $p_0^*(\boldsymbol{x}) = 1$. Then we define for $\epsilon > 0$

$$q_k^*(\boldsymbol{x}) := \frac{p_k^*(\boldsymbol{x}) + \epsilon}{p_{k-1}^*(\boldsymbol{x}) + 2\epsilon}, \quad k = 1, \ldots, K-1, \tag{5}$$

and note that for each k and $\boldsymbol{x}$ the function $q_k(\boldsymbol{x})$ is continuous and satisfies

$$0 < q_k^*(\boldsymbol{x}) < 1.$$

Then define the continuous functions

$$a_k^*(\boldsymbol{x}) := \sigma^{-1}(q_k^*(\boldsymbol{x})) = \log \frac{q_k^*(\boldsymbol{x})}{1 - q_k^*(\boldsymbol{x})}, \quad k = 1, \ldots, K-1.$$

Because the upstream neural network can approximate any continuous function $g : \mathbb{X} \to \mathbb{R}^{K-1}$ with error $\mathcal{O}(\epsilon)$, we can set $g_k(\boldsymbol{x}) = a_k^*(\boldsymbol{x})$ and have the upstream neural network produce the functions $\widehat{g}_k(\boldsymbol{x}) = a_k^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)$ for all k and $\boldsymbol{x}$. Then after the CONDOR sigmoid activations we would have the neural network produce for all $\boldsymbol{x}$ at its output nodes[1]

$$\sigma(a_k^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)), \quad k = 1, \ldots, K-1,$$

which by Taylor Series about $a_k^*(\boldsymbol{x})$ produces

$$\begin{aligned}
\sigma(a_k^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)) &= \sigma(a_k^*(\boldsymbol{x})) + \sigma'(a_k^*(\boldsymbol{x}))\mathcal{O}(\epsilon) + \sigma''(a_k^*(\boldsymbol{x}))\mathcal{O}(\epsilon^2)/2 + \ldots \\
&= \sigma(a_k^*(\boldsymbol{x})) + \sigma(a_k^*(\boldsymbol{x}))(1 - \sigma(a_k^*(\boldsymbol{x})))\mathcal{O}(\epsilon) + \mathcal{O}(\epsilon^2) \\
&= \sigma(a_k^*(\boldsymbol{x})) + \mathcal{O}(\epsilon) \\
&= q_k^*(\boldsymbol{x}) + \mathcal{O}(\epsilon).
\end{aligned}$$

Then the CONDOR approach yields

$$\begin{aligned}
p_k(\boldsymbol{x}) &= \prod_{k'=1}^{k} \sigma(a_k^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)) = \prod_{k'=1}^{k} [q_{k'}^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)] \\
&= \mathcal{O}(\epsilon) + q_1^*(\boldsymbol{x}) \prod_{k'=2}^{k} [q_{k'}^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)] \\
&= \mathcal{O}(\epsilon) + q_1^*(\boldsymbol{x}) q_2^*(\boldsymbol{x}) \prod_{k'=3}^{k} [q_{k'}^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)].
\end{aligned}$$

[1] Note that only the activation at the last layer is fixed as the sigmoid function. The activation of the hidden layers could be other functions, e.g., ReLU.

By iteration it follows that

$$p_k(\boldsymbol{x}) = \left[\prod_{k'=1}^{k} q_{k'}^*(\boldsymbol{x})\right] + \mathcal{O}(\epsilon).$$

Using the definition of $q_k^*(\cdot)$ in equation 5, we get

$$\begin{aligned}
p_k(\boldsymbol{x}) &= \left[\frac{\prod_{k'=1}^{k}[p_{k'}^*(\boldsymbol{x})+\epsilon]}{\prod_{k'=1}^{k}[p_{k'-1}^*(\boldsymbol{x})+2\epsilon]}\right] + \mathcal{O}(\epsilon) \\
&= \frac{[\prod_{k'=1}^{k} p_{k'}^*(\boldsymbol{x})] + \mathcal{O}(\epsilon)}{[\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x})] + \mathcal{O}(\epsilon)} + \mathcal{O}(\epsilon) \\
&= \frac{\prod_{k'=1}^{k} p_{k'}^*(\boldsymbol{x})}{[\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x})] + \mathcal{O}(\epsilon)} + \frac{\mathcal{O}(\epsilon)}{[\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x})] + \mathcal{O}(\epsilon)} + \mathcal{O}(\epsilon) \\
&= \frac{\prod_{k'=1}^{k} p_{k'}^*(\boldsymbol{x})}{[\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x})] + \mathcal{O}(\epsilon)} + \mathcal{O}(\epsilon) \qquad (6) \\
&= p_k^*(\boldsymbol{x}) + \mathcal{O}(\epsilon) \qquad (7)
\end{aligned}$$

for all $\boldsymbol{x}$ and $k = 1, \ldots, K-1$. The last Equality (7) comes from considering separately the cases $\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x}) = 0$ and $\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x}) > 0$. In the former we have $p_k^*(\boldsymbol{x}) = 0$ and Equation (6) reduces to $p_k(\boldsymbol{x}) = 0 + \mathcal{O}(\epsilon)$. In the latter we find from Equation (6)

$$\begin{aligned}
p_k(\boldsymbol{x}) &= \frac{1/\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x})}{1/\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x})} \times \frac{[\prod_{k'=1}^{k} p_{k'}^*(\boldsymbol{x})]}{[\prod_{k'=1}^{k} p_{k'-1}^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)]} + \mathcal{O}(\epsilon) \\
&= \frac{p_k^*(\boldsymbol{x})}{[1+\mathcal{O}(\epsilon)]} + \mathcal{O}(\epsilon) \\
&= p_k^*(\boldsymbol{x})[1+\mathcal{O}(\epsilon)] + \mathcal{O}(\epsilon) \qquad (8) \\
&= p_k^*(\boldsymbol{x}) + \mathcal{O}(\epsilon)
\end{aligned}$$

where Equality (8) comes from the power series $1/(1+x) = 1 - x + x^2 - \ldots$. □

3 Numerical Experiments

Here we discuss the challenges of measuring the performance of ordinal regression methods as well as the potential strengths and weaknesses of various candidate measures. In the subsequent sections, we then demonstrate the superiority of the performance of CONDOR compared to the state-of-the-art on several benchmark and real-world data sets.

The binary cross entropy (BCE) loss in Equation (4) is one of the few common evaluation metrics fully sensitive to the ordinal nature of the outcomes, since most other metrics either ignore crucial information about ordering (e.g., accuracy, which is categorical and thus lacks any notion of "incorrect but close") or artificially impose an interval or ratio scale on the data (e.g., mean absolute error or earth movers distance, which by necessity require metric distances to be defined between each rank). Despite its suitability, the cross entropy remains a less intuitive method than its alternatives and so we benchmark these additional performance measures while acknowledging their shortcomings in the ordinal setting. Specifically, we profile the earth movers distance (EMD) on the rank indices with unit distance between ranks, and the mean absolute error (MAE) on the rank indices with unit distance between ranks using Equation (1) for the point estimate.

In the subsequent sections, the only difference between the three methods is the choice of loss function and final layer of the neural network; all other details of the DNN architecture, the optimization algorithms, hyperparameters and random number seeds are kept equal throughout each experiment. Namely, CORAL and CONDOR both have the sum of the BCE for the $K-1$ subtasks as a loss, whereas categorical uses the CCE. Likewise, CORAL uses a custom final layer with weight sharing Cao et al. (2020) among $K-1$ output nodes, CONDOR uses a final dense layer with $K-1$

output nodes which after sigmoid activation represent $q_k(\boldsymbol{x}), k = 1, \ldots, K-1$, and the categorical algorithm uses a dense layer with K nodes and a softmax activation. All results were gathered with three random number seeds and reported as the mean plus or minus the standard deviation across these seeds.

3.1 Synthetic Quadrants Dataset

We consider the simple task of ordinal classification wherein the labels $0, 1, 2, 3$ are the quadrants of the plane going counterclockwise and the features are generated from a 2D standard normal distribution. We draw 1000 samples and do a 90/10 train/test split of the dataset. We select as the upstream network architecture two dense layers with ten neurons and RELU activations and an Adam optimizer with 100 epochs and early stopping patience of 10 using a validation split of 0.2. As can be seen in Table 1, CONDOR has the best performance in BCE, EMD and MAE.

Table 1: Synthetic quadrants data results in test set

ALGORITHM	**BCE**	**MAE**	**EMD**
CONDOR	**0.0768 ± 0.0100**	**0.0167 ± 0.0153**	**0.0799 ± 0.0526**
CORAL	0.5074 ± 0.0754	0.0733 ± 0.0416	0.4080 ± 0.0377
CATEGORICAL	1.3438 ± 0.0458	0.0200 ± 0.0173	1.0318 ± 0.0267

3.2 MNIST as an ordinal dataset

Depending on the application, MNIST can be considered a categorical problem or an interval regression problem. If the digits are used for license-plate recognition, then the problem is categorical since there is no notion of "close" errors. By contrast, if the digits are used for GPS coordinates or postal codes, then the ordering and distance between numerals becomes relevant and categorical classification is no longer the most appropriate framing of the task. It is valid to treat interval regression as an ordinal problem since the latter assumes less structure on the dataset, although it is recommended to exploit the interval scale. Here we treat MNIST as ordinal data for the purpose of benchmarking our ordinal algorithms, while acknowledging that it should likely be treated as either a categorical classification or interval regression as dictated by the specific real-world application setting. The MNIST data are split into training, validation and test sets of 55K, 5K and 10K images, respectively. We utilize a convolutional neural network with two convolutional layers of 64 and 32 filters respectively and a kernel size of 3, before flattening and passing to the appropriate output layer and loss function for our three models (CONDOR, CORAL, Categorical). Training is performed with the Adam optimizer, a maximum of 100 epochs and an early-stopping patience of 10. The results in Table 2 indicate that CONDOR demonstrates superior performance in terms of both BCE and EMD, while the Categorical approach performs marginally better in MAE.

Table 2: MNIST digit image classification on test set

ALGORITHM	**BCE**	**MAE**	**EMD**
CONDOR	**0.1784 ± 0.0043**	0.0596 ± 0.0027	**0.0818 ± 0.0065**
CORAL	1.2724 ± 0.0139	0.4583 ± 0.0028	0.7501 ± 0.0214
CATEGORICAL	5.5424 ± 0.0013	**0.0592 ± 0.0042**	3.0638 ± 0.0013

3.3 NLP on Amazon reviews dataset

Here we consider a natural language processing (NLP) dataset consisting of 99,025 (non-duplicate and non-empty) Amazon Pantry text reviews with their corresponding one to five star ratings (Ni et al., 2019). We split the data to have a test set with 10,000 examples. For the neural network architecture, we use the fixed and pre-trained Google universal sentence encoder (Cer et al., 2018) and append a dense layer with 64 ReLu-activated neurons and a dropout of 0.1, followed by the appropriate output layer and loss function for each model. Training is performed with the Adam optimizer, a maximum of 100 epochs and an early-stopping patience of 10 with a validation split of 0.2. The results in Table 3 demonstrate that CONDOR provides the strongest performance in this benchmark across all three performance metrics.

Table 3: Amazon product review text classification on test set

ALGORITHM	**BCE**	**MAE**	**EMD**
CONDOR	**0.7807 ± 0.0113**	**0.3180 ± 0.0047**	**0.4582 ± 0.0050**
CORAL	0.8095 ± 0.0098	0.3263 ± 0.0052	0.4726 ± 0.0041
CATEGORICAL	2.4663 ± 0.4678	0.4195 ± 0.0074	1.6906 ± 0.2422

3.4 GRU-D FOR COVID-19 PROGNOSTICATION

This study adheres to a research protocol approved by the [CENSORED FOR DOUBLE BLIND PEER REVIEW]Institutional Review Board. Here we extend the results from Sankaranarayanan et al. (2021) to progress from their binary classification predicting mortality to ordinal regression predicting severity of outcomes. Namely, this clinical dataset includes two binary severity outcomes: an indicator variable for mechanical ventilation or extracorporeal membrane oxygenation (ECMO), as well as an indicator variable of whether patient death occurred. From these, a clear three point ordinal scale can be constructed whereby a patient is scored a zero when they have no severe outcome, a one when they experienced the severe outcome of ventilation or ECMO, and a two corresponding to death (with or without prior ventilation or ECMO). Sankaranarayanan et al. (2021) identified the GRU-D recurrent neural network architecture as the best performing model for binary mortality prediction, and we extend that approach to the ordinal problem. We do this for the CONDOR, CORAL, and categorical algorithms using their corresponding final layers and loss functions. This GRU-D architecture (Che et al., 2018) deals explicitly with the 55 dimensional time series that is missing not at random (MNAR) due to the manner in which clinical data is ordered and recorded in an electronic health record (EHR). The default hyperparameters (dropout of 0.3, l2 regularizaton of 0.0001, 100 hidden and recurent neurons, batch size of 256, adam learning rate 0.001, no batch norm, no bidirectional RNN, 50 max time steps, 100 epochs with early stopping patience of 10 epochs) have previously demonstrated strong performance (Sankaranarayanan et al., 2021) and so are retained here.

Table 4: COVID-19 severity prediction results in prospective test set

ALGORITHM	**BCE**	**MAE**	**EMD**
CONDOR	**0.6526 ± 0.0004**	**0.2986 ± 0.0044**	**0.4151 ± 0.0063**
CORAL	0.6711 ± 0.0000	0.3021 ± 0.0015	0.4261 ± 0.0023
CATEGORICAL	1.1075 ± 0.0019	0.3076 ± 0.0021	0.8185 ± 0.0010

The dataset is split into a training/validation set of 9,435 patients who tested positive for COVID-19 prior to December 15 2020 by PCR test, and a prospective testing set of 2,372 patients who tested positive on or after that date. For training we use an identical 90/10 training/validation split on the 9,435, which facilitates early stopping with patience. The results in Table 4 demonstrate that CONDOR is superior in all of evaluation metrics. Furthermore, the CONDOR-based GRU-D model has a prospective test set AUROC of 0.9038 ± 0.0025 for the mortality prediction subtask, which is greater than the 0.901 reported in Sankaranarayanan et al. (2021) wherein the authors trained the algorithm as a binary classifier specifically for mortality prediction. This demonstrates that there is no loss of mortality prediction performance when building a DNN to address the more challenging task of prognostication.

4 Discussion

We have demonstrated the ability of the CONDOR approach to overcome limitations present in popular alternative methods and to produce rank consistent results in the classification of data with ordinal labels. Rank consistency is not only important for theoretical soundness, but in application settings where explainability is important and a rank inconsistent prediction will be unacceptably contradictory and fundamentally unexplainable. Regardless of the loss function being optimized or the parameterization of the neural network, CONDOR provides universal guarantees of rank consistency by Lemma 2.1, which is to say the CONDOR approach is “sufficient” for rank consistency. Our next result leverages the fact that there are a wide-variety of universal approximation theorems for neural networks each with their own technical conditions (e.g., see Chong (2020) for discussion of various universal approximation proofs and technical conditions). Namely, Theorem 2.2 states than any upstream neural network satisfying the conditions for universal approximation can be provided a CONDOR output layer, which will create a universally rank consistent network that can approximate any rank consistent solution. This theorem can be interpreted as CONDOR being “necessary” for rank consistency, insofar as any rank consistent solution can be represented by a CONDOR neural network.

In contrast, note that Theorem 1 of Cao et al. (2020) only provides rank consistency at the global minimum of an optimization problem with the specified loss function. Since neural network training is not guaranteed or expected to reach a global optimal parameterization, the Cao et al. (2020) approach in practice may produce rank inconsistent solutions and requires *post hoc* checks of the estimated bias terms to verify rank consistency. Furthermore, Cao et al. (2020) restricts expressiveness in the $K - 1$ binary classifier outputs that must have “parallel slopes” (i.e., differ only by a bias parameter whose impact is completely independent of the feature vector). In Appendix A.1, we formalize these comments with two proofs demonstrating that the CORAL framework (Cao et al., 2020) lacks the theoretical guarantees of CONDOR.

Beyond our mathematical justifications, ultimately it is critical that the method perform well within a wide variety of neural network architectures and ordinal problem settings. Our benchmarking of dense networks, CNNs, attention networks, and exotic RNNs all shows practical benefit of using the CONDOR algorithm in a diverse set of applications. In all benchmarks, CONDOR provided the best performance according to the ordinal metric of BCE. Furthermore, CONDOR often provided the best performance according to the interval-scale metrics of EMD and MAE. We emphasize however that both of these metrics assume a interval scale on the ranks (i.e., uniform-spacing between ranks) that is not formally justified from a mathematical perspective in an ordinal setting. Less formally, if we recall the ordinal example, “terrible”<“great”<“best”, then uniform spacing would not be compatible with most readers’ intuition about the problem. Any attempt to arbitrarily embed these three ranks into a metric space would be unlikely to achieve universal agreement amongst practitioners. Thus, ordinal regression is best exemplified in use-cases where a well-defined ordering is clear but distances between the ranks are undefined. In that sense the BCE is the only metric that directly evaluates ordinal performance. Nonetheless CONDOR not only remains competitive in the categorical performance measure of accuracy, but in-fact provides improved classification in true ordinal problems when compared to categorically optimized neural networks, in theory due to the ability of the network to exploit “clues” encountered during ordinal training (Appendix A.2).

In addition to the theoretical strengths and performance improvements of the CONDOR method, we note that most applied papers simply use categorical classification in their problem settings rather than consider current state-of-the-art methods for ordinal regression. Part of this may be educational, as most beginners are only taught binary/categorical classification and continuous regression, and discussions of the ordinal setting are largely only found in the advanced literature. However, the authors also believe part of the barrier is programmatic ease-of-use. We provide production-ready and user-friendly software packages in both PyTorch and TensorFlow, in order to minimize the effort required to convert existing categorical code-bases into CONDOR ordinal code-bases. In Appendix A.3, we demonstrate the modest code changes required to implement our methodology in an existing categorical code base.

The key requirements for successful supervised learning tasks are algorithms that respect the structure of the problem, and access to sufficient amounts of labeled data. Since CONDOR satisfies the first requirement by providing a robust algorithm for ordinal regression, we conclude with the latter by emphasizing the prevalence of available ordinal outcome measurements, using medical applications as a prototypical applied problem domain. Survey research for instance, frequently utilizes ordinal responses such as the psychometric Likert scale (Likert, 1932), providing a large corpus of existing data with ordinal labels. Furthermore, while labeling outcomes from the Electronic Health Record (EHR) is one of the most time-consuming and expensive aspects of applied machine learning in the medical space, the proliferation of ordinal scales in modern medical practice (see Appendix A.4) means the EHR already contains physician-provided ordinal outcomes from a large variety of settings. The ubiquity of ordinal outcome measurements throughout survey research and medical settings represents a rich untapped reserve of training data that have yet to be explored by ordinal regression algorithms, and it is our hope that CONDOR's demonstrated capabilities and its ease-of-use will encourage its adoption and enable broad exploration of underutilized data across these and other domains.

Reproducibility Statement

The PyTorch and TensorFlow repositories implementing CONDOR will be made available on GitHub under open source licenses, following completion of double-blind peer review. Anonymized versions of these packages (i.e., we have removed/scrambled identifying links to GitHub or authorship details, and all remaining links are to repositories for competing methods that are not our own) have been provided as supplemental materials with this submission to facilitate peer-review. All hyperparameters and preprocessing steps have been provided in the corresponding numerical results sections. Three of the four data sets are publicly available and the supplemental code provided to reviewers includes scripts to enable reproduction of our findings in these data sets. The COVID-19 dataset contains Protected Health Information subject to Institutional Review Board oversight that prohibits public release at this time.

Acknowledgments

The funding for this research has been provided by the [CENSORED FOR BLIND REVIEW]. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Evaluation measures for ordinal regression. In *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pp. 283–287, 2009. doi: 10.1109/ISDA.2009.230.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331, dec 2020. doi: 10.1016/j.patrec.2020.11.008. URL `https://doi.org/10.1016%2Fj.patrec.2020.11.008`.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *arXiv*, pp. 1803.11175, 2018. URL `https://arxiv.org/abs/1803.11175`.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

Kai Fong Ernest Chong. A closer look at the approximation capabilities of neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rkevSgrtPr`.

Wei Chu and S. Sathiya Keerthi. New approaches to support vector ordinal regression. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 145–152, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102370. URL `https://doi.org/10.1145/1102351.1102370`.

K. Crammer and Y. Singer. Pranking with ranking. *Advances in Neural Information Processing Systems*, 14:641–647, 2002. cited By 207.

M. Forrest and B. Andersen. Ordinal scale and statistics in medical research. *Br Med J (Clin Res Ed)*, 292(6519):537–538, Feb 1986.

M. C. B. Godoy, E. G. L. C. Odisio, J. J. Erasmus, R. C. Chate, R. S. Dos Santos, and M. T. Truong. Understanding Lung-RADS 1.0: A Case-Based Review. *Semin Ultrasound CT MR*, 39(3):260–272, Jun 2018.

Gil Levi and Tal Hassncer. Age and gender classification using convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 34–42, 2015. doi: 10.1109/CVPRW.2015.7301352.

L. Li and H.-T. Lin. Ordinal regression by extended binary classification. *Advances in Neural Information Processing Systems*, pp. 865–872, 2007.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140):55, 1932.

P. McCullagh. Regression models for ordinal data (with discussion). *Journal of the Royal Statistical Society, Series B*, 42:109–142, 1980. cited By 12.

C. L. Mercado. BI-RADS update. *Radiol Clin North Am*, 52(3):481–487, May 2014.

E. E. Moore and F. A. Moore. American Association for the Surgery of Trauma Organ Injury Scaling: 50th anniversary review article of the Journal of Trauma. *J Trauma*, 69(6):1600–1601, Dec 2010.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fined-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 188–197, 2019.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4920–4928, 2016. doi: 10.1109/CVPR.2016.532.

K. Obermayer. Support vector learning for ordinal regression. *IET Conference Proceedings*, pp. 97–102(5), 1999. URL `https://digital-library.theiet.org/content/conferences/10.1049/cp_19991091`.

Shyamsundar Rajaram, Ashutosh Garg, Xiang Sean Zhou, and Thomas S. Huang. Classification approach towards ranking and sorting problems. In Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski (eds.), *Machine Learning: ECML 2003*, pp. 301–312, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-39857-8.

F. C. M. Reith, H. F. Lingsma, B. J. Gabbe, F. E. Lecky, I. Roberts, and A. I. R. Maas. Differential effects of the Glasgow Coma Scale Score and its Components: An analysis of 54,069 patients with traumatic brain injury. *Injury*, 48(9):1932–1943, Sep 2017.

S. Richards, N. Aziz, S. Bale, D. Bick, S. Das, J. Gastier-Foster, W. W. Grody, M. Hegde, E. Lyon, E. Spector, K. Voelkerding, and H. L. Rehm. Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. *Genet Med*, 17(5):405–424, May 2015.

Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 252–257, 2015. doi: 10.1109/ICCVW.2015.41.

Saranya Sankaranarayanan, Jagadheshwar Balan, Jesse R Walsh, Yanhong Wu, Sara Minnich, Amy Piazza, Collin Osborne, Gavin R Oliver, Jessica Lesko, Kathy L Bates, Kia Khezeli, Darci R Block, Margaret DiGuardo, Justin Kreuter, John C O'Horo, John Kalantari, Eric W Klee, Mohamed E Salama, Benjamin Kipp, William G Morice, and Garrett Jenkinson. Covid-19 mortality prediction from deep learning in a large multistate electronic health record and laboratory information system data set: Algorithm development and validation. *J Med Internet Res*, 23(9):e30157, Sep 2021. ISSN 1438-8871. doi: 10.2196/30157. URL https://www.jmir.org/2021/9/e30157.

Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pp. 961–968, Cambridge, MA, USA, 2002. MIT Press.

Libin Shen and Aravind K. Joshi. Ranking and reranking with perceptron. *Machine Learning*, 60(1-3):73–96, jun 2005. doi: 10.1007/s10994-005-0918-9. URL https://doi.org/10.1007%2Fs10994-005-0918-9.

S. G. Silverman, I. Pedrosa, J. H. Ellis, N. M. Hindman, N. Schieda, A. D. Smith, E. M. Remer, A. B. Shinagare, N. E. Curci, S. S. Raman, S. A. Wells, S. D. Kaffenberger, Z. J. Wang, H. Chandarana, and M. S. Davenport. Bosniak Classification of Cystic Renal Masses, Version 2019: An Update Proposal and Needs Assessment. *Radiology*, 292(2):475–488, 08 2019.

A. Williamson and B. Hoggart. Pain: a review of three commonly used pain rating scales. *J Clin Nurs*, 14(7):798–804, Aug 2005.

S. Zachariah, W. Wykes, and D. Yorston. Grading diabetic retinopathy (DR) using the Scottish grading protocol. *Community Eye Health*, 28(92):72–73, 2015.

A Supplemental Appendix

A.1 CORAL proofs

In this appendix, we demonstrate formally that the CORAL framework of Cao et al. (2020) does not have the universal rank consistency nor the expressiveness of CONDOR.

Lemma A.1. *CORAL is not universally rank consistent.*

Proof. For CORAL the last layer shares weights and only has different bias terms $b_1, \dots, b_k$ meaning it represents the output probabilities as (Cao et al., 2020)

$$p_k(\boldsymbol{x}) = \sigma(a(\boldsymbol{x}) + b_k) = \frac{1}{1 + \exp(-a(\boldsymbol{x}))\exp(-b_k)} \quad k = 1, \dots, K-1,$$

for some $a(\boldsymbol{x})$. This means in the notation of CONDOR that

$$\begin{aligned} q_k(\boldsymbol{x}) &= \frac{p_k(\boldsymbol{x})}{p_{k-1}(\boldsymbol{x})} = \frac{1 + \exp(-a(\boldsymbol{x}))\exp(-b_{k-1})}{1 + \exp(-a(\boldsymbol{x}))\exp(-b_k)} \\ &= \frac{\exp(a(\boldsymbol{x})) + \exp(-b_{k-1})}{\exp(a(\boldsymbol{x})) + \exp(-b_k)} \end{aligned} \tag{9}$$

for $k = 2, \dots, K-1$ and $q_1(\boldsymbol{x}) = p_1(\boldsymbol{x})$. Note that if the neural network parameters are chosen such that $b_k > b_{k-1}$ for some k then $q_k(\boldsymbol{x}) > 1$ and CORAL is rank inconsistent. □

It is quite clear from Equation (9) that the functional form of CORAL is far more restrictive than CONDOR, which allows arbitrary $q_k : \mathbb{X} \to [0, 1]$ functions. For completeness, we prove formally in the next lemma that it is less expressive.

Lemma A.2. *CORAL can not approximate every rank consistent solution with $\mathcal{O}(\epsilon)$ error.*

Proof. For simplicity, consider a univariate input x and $K = 3$. Then we have for CORAL

$$\begin{aligned} q_2(x) &= \frac{p_2(x)}{p_1(x)} = \frac{1 + \exp(-a(x))\exp(-b_1)}{1 + \exp(-a(x))\exp(-b_2)} \\ &= \frac{\exp(a(x)) + \exp(-b_1)}{\exp(a(x)) + \exp(-b_2)}, \end{aligned}$$

and $q_1(x) = p_1(x) = \sigma(a(x) + b_1)$. Consider an extremely simple CONDOR network with no hidden layers, bias parameters fixed to zero and only two weights $w_1 = 1, w_2 = 2$ producing

$$q_k^*(x) = \frac{1}{1 + \exp(kx)}, \quad k = 1, 2$$

which is rank consistent by Lemma 2.1. Suppose by way of contradiction that there exists CORAL $a(x)$ and b_1, b_2 such that $q_k(x) = q_k^*(x) + \mathcal{O}(\epsilon)$ for all k and x. Thus

$$\begin{aligned} \frac{1}{1 + \exp(-a(x))\exp(-b_1)} &= \frac{1}{1 + \exp(x)} + \mathcal{O}(\epsilon) \\ \frac{\exp(a(x)) + \exp(-b_1)}{\exp(a(x)) + \exp(-b_2)} &= \frac{1}{1 + \exp(2x)} + \mathcal{O}(\epsilon) \end{aligned}$$

which implies

$$\exp(a(x)) = \exp(-x - b_1) + \mathcal{O}(\epsilon) \tag{10}$$

$$(\exp(a(x)) + \exp(-b_1))(1 + \exp(2x)) = \exp(a(x)) + \exp(-b_2) + \mathcal{O}(\epsilon) \tag{11}$$

and plugging in Equation (10) into Equation (11) we find after rearrangement

$$\exp(x) + \exp(2x) = \exp(b_1 - b_2) - 1 + \mathcal{O}(\epsilon)$$

which is a contradiction since the left hand side has an infinite range depending on x whereas the right hand side is a constant up to an error of order ϵ.

□

A.2 CATEGORICAL ACCURACY RESULTS

Accuracy is a categorical performance measure wherein there is not an increasing penalty for being further from the correct label, and therefore no relative "credit" given for being close to the correct label. One might therefore expect that training a neural network with a categorical loss (i.e., CCE) would result in higher categorical accuracy than if the network were trained with a ordinal method.

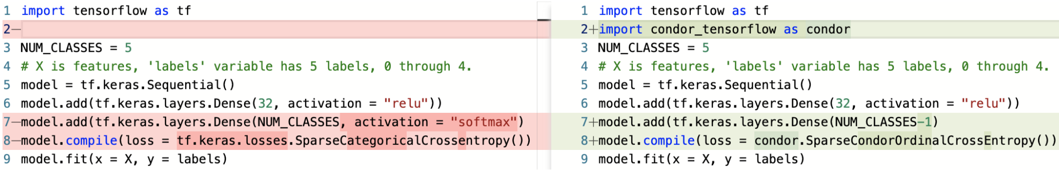
Table 5: Accuracy in benchmark test sets

ALGORITHM	Quadrants	MNIST	Amazon	COVID-19
CONDOR	0.9900 ± 0.0100	0.9805 ± 0.0004	**0.7498 ± 0.0032**	**0.7250 ± 0.0033**
CORAL	0.9333 ± 0.0379	0.6381 ± 0.0037	0.7343 ± 0.0036	0.7175 ± 0.0015
CATEGORICAL	**0.9933 ± 0.0058**	**0.9845 ± 0.0008**	0.7299 ± 0.0037	0.7244 ± 0.0016

Surprisingly, in Table 5 we find in some benchmarks the ordinal CONDOR method provided higher categorical accuracy than the networks trained using categorical cross entropy to specifically optimize categorical performance. We attribute this remarkable finding to the "clues" provided to the network when the ordinal nature of the problem is exploited during training. For instance, if the network incorrectly guesses rank index 7 when the true rank index is 8, the categorical loss treats this equivalently to a guess of a rank index of 1; the back propagation does not send any signals indicating that the guess of rank 7 was "close" to the true rank of 8. By contrast, Equation (4) would capture the fact that most of the binary subtasks are correctly predicted when a rank 7 is estimated for a ground truth rank of 8. In problems like MNIST where the features are not necessarily trending with increasing rank, we could understand how a categorical loss produces a stronger categorical accuracy. But in problems like Amazon star ratings, where the language and sentiment of 4 and 5 star reviews are likely closer in the NLP embedding space than the language and sentiment of 1 star reviews, one can also understand how training with an ordinal loss could provide higher categorical accuracy than a categorical loss that only has the capacity to indicate “correct” versus “incorrect” and never “incorrect but close”.

A.3 Minimal code changes

CONDOR is open sourced as both TensorFlow ([GITHUB LINK BLINDED FOR PEER REVIEW, zip file provided to reviewers]) and PyTorch ([GITHUB LINK BLINDED FOR PEER REVIEW, zip file provided to reviewers]) repositories that make it simple to modifying existing categorical code bases to use CONDOR. See Figure 2 for a hypothetical example in TensorFlow. Both the TensorFlow and PyTorch versions of the GitHub repositories have full mkdocs documentation, docker files, ipynb tutorials, continuous integration testing and pip packaging. The authors hope this reduces the barrier to using proper and cutting-edge ordinal regression in applied problems.



```
 import tensorflow as tf
-
 NUM_CLASSES = 5
 # X is features, 'labels' variable has 5 labels, 0 through 4.
 model = tf.keras.Sequential()
 model.add(tf.keras.layers.Dense(32, activation = "relu"))
-model.add(tf.keras.layers.Dense(NUM_CLASSES, activation = "softmax"))
-model.compile(loss = tf.keras.losses.SparseCategoricalCrossentropy())
 model.fit(x = X, y = labels)
```

```
 import tensorflow as tf
+import condor_tensorflow as condor
 NUM_CLASSES = 5
 # X is features, 'labels' variable has 5 labels, 0 through 4.
 model = tf.keras.Sequential()
 model.add(tf.keras.layers.Dense(32, activation = "relu"))
+model.add(tf.keras.layers.Dense(NUM_CLASSES-1))
+model.compile(loss = condor.SparseCondorOrdinalCrossEntropy())
 model.fit(x = X, y = labels)
```

Figure 2: Converting a categorical code base into a CONDOR ordinal code base is straightforward.

A.4 Ordinal outcomes in medical practice

Modern medical practice emphasizes the importance of a standardized and reproducible communication of findings and outcomes within the global medical community. Consistent diagnosis, prognostication, treatment, and decision making all require evidence- and consensus-based labeling of patient disease states. Frequently, these categorizations are made ordinal to align with the expected prognosis or severity of disease.

As a result, across nearly every sub-specialty of medicine, one can find a plethora of ordinal outcome scales. Well-known to the general public is the use of tumor staging in oncology to characterize neoplasms. However, we provide a non-comprehensive sampling of other specialties that are perhaps less well-known. For instance, the American Association for the Surgery of Trauma provides 32 ordinal scales (Moore & Moore, 2010) for assessing the severity of trauma to 32 organs on scale of 1 (minimal) to 6 (lethal). Molecular testing results, such as DNA variant sequencing, are often graded on an ordinal scale such as The American College of Medical Genetics and Genomics’ variant scoring from variants from benign, likely benign, variant of unknown significance, likely pathogenic, to pathogenic (Richards et al., 2015). Subjective outcomes are often measured on ordinal scales, such as the 11-level Pain Rating Scale (Williamson & Hoggart, 2005) from 0 (no pain) to 10 (worst possible pain). Traumatic brain injury is assessed on the 5-point Glasgow outcome scale (Reith et al., 2017). Radiologists make frequent use of ordinal scales, including but not limited to Lung-RADS screening lesions from 0 to 4 (Godoy et al., 2018), BI-RADS for breast cancer screening from 0 to 6 (Mercado, 2014), and Bozniak classifcation for renal lesions from 1 to 4 (Silverman et al., 2019). Retinopathy has the Scottish Grading protocol from 0 to 4 (Zachariah et al., 2015). While these examples are not intended to be a comprehensive review, they hopefully provide some insight into just how prevalent ordinal scales are in modern medicine.