Pattern Recognition 122 (2022) 108310

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Unimodal regularisation based on beta distribution for deep ordinal regression

Víctor Manuel Vargas*, Pedro Antonio Gutiérrez, César Hervás-Martínez

Department of Computer Science and Numerical Analysis, University of Córdoba, Campus Universitario de Rabanales, "Albert Einstein" building, 3rd floor. 14014 Córdoba, Spain

ARTICLE INFO

Article history:
Received 3 March 2021
Revised 6 September 2021
Accepted 7 September 2021
Available online 10 September 2021

Keywords:
Ordinal regression
Unimodal distribution
Convolutional network
Beta distribution
Stick-breaking

ABSTRACT

Currently, the use of deep learning for solving ordinal classification problems, where categories follow a natural order, has not received much attention. In this paper, we propose an unimodal regularisation based on the beta distribution applied to the cross-entropy loss. This regularisation encourages the distribution of the labels to be a soft unimodal distribution, more appropriate for ordinal problems. Given that the beta distribution has two parameters that must be adjusted, a method to automatically determine them is proposed. The regularised loss function is used to train a deep neural network model with an ordinal scheme in the output layer. The results obtained are statistically analysed and show that the combination of these methods increases the performance in ordinal problems. Moreover, the proposed beta distribution performs better than other distributions proposed in previous works, achieving also a reduced computational cost.

1. Introduction

In the last decade, ordinal classification/regression has received an increasing interest in the literature [1–3]. The methods focused on solving this kind of problems aim to determine the discrete category or ranking of a pattern in an ordinal scale. This ordinal scale is given by the natural ordering of the categories existing in the problem considered [4]. For instance, in medical problems where we obtain a diagnosis from images, the category is usually in an ordinal scale (e.g. Diabetic Retinopathy (DR) detection [5] with five levels of the disease). Another possible example is the prediction of the age range of people from photographs of their faces [6].

There are many real world problems where the available data have an underlying ordinal structure. In [7], the authors aim to predict the level of the Parkinson's disease based on volumetric images obtained through encephalograms. The patients are classified depending on the state of this pathology (1: healthy patient, 2: slight alteration, 3: more advanced alteration, etc.). In [8] the authors try to predict convective situations in the Madrid-Barajas airport in Spain, which is crucial for this kind of transportation facilities as it can cause severe impact in flight scheduling and safety. These situations can be present in several degrees, resulting in different classes that follow a natural order. Finally, in [9] authors try to automatically detect prostate cancer on different degrees based on the Gleason Score, which is a standard for measuring the aggressiveness of this type of cancer. According to this metric, prostate cancer is divided into 5 categories, where the first one does not require a treatment while the others do. Also, depending on the aggressiveness degree, the treatment must be different, and it is important to determine this level accurately to avoid excessive or insufficient treatments.

In any of these real world problems, misclassifying a pattern in an adjacent class is always less important than misclassifying it in distant classes. This is the main reason why taking the ordinal information into account when solving this kind of problems is essential. Moreover, when ordinal classifiers are used, the order of the labels is explicitly considered in the model, what generally accelerates the learning process and reduces the amount of data needed for training.

Machine learning methods based on Deep Learning [10] have been used for a wide variety of tasks. Deep Neural Networks (DNN) have the ability to obtain high level abstract representations of low level features. Each layer of the network extract higher level features from the previous layer. Specifically, Convolutional Neural Networks (CNN) take an image on gray-scale or RGB colour as input data and extract a set of features that are used to classify the pattern in one of the different categories or rankings. This kind of models have been used in problems related with image classifi-

* Corresponding author at: Campus Universitario de Rabanales, "Albert Einstein" building, 3rd floor. 14014 Córdoba, Spain.
E-mail address: vvargas@uco.es (V. Manuel Vargas).

https://doi.org/10.1016/j.patcog.2021.108310

cation [11], speech recognition [12], control problems [13], object detection [14], privacy and security protection [15], recovery of human pose [16], semantic segmentation [17], image retrieval [18,19], visual recognition [20], etc.

However, the resolution of ordinal problems with deep learning models has not received much attention. The most common approach to solve ordinal data problems is to treat them as multi-class problems and optimise the model using the standard cross-entropy (CE) loss [21]. This approach has a major drawback: it does not takes into account the order between categories. Some previous works [22,23] have explored different loss functions to address this problem. Another common approach is to convert the ordinal regression problem to a standard regression one [24]. This approach keeps the order between rankings but assumes that the discrete categories are continuous and equispaced.

Another problem when working with ordinal data is found in the way the labels are encoded. Usually, the one-hot encoding is used, which represents the label as binary vector where the j element is 1 when the true class is j. However, the way this encoding represents the labels incurs in the same penalty for all misclassification errors, without taking into account the distance to the true label. A better approach for ordinal regression problems is to use a smooth label representation, in such a way that classes which are close to the real label produce a smaller error than classes that are far. This method, known as label smoothing or unimodal regularisation for the loss function, has been used in previous works and different distribution functions have been used to model the shape of these smooth labels (poisson, binomial [25] or exponential [26]).

In this work, we propose to use beta distributions for the label smoothing method together with an approach to automatically determine the parameters of these distributions. To evaluate the performance of the proposed method, we use the unimodal regularised loss to train a CNN model with three separate images datasets. The unimodal regularisation is combined with the recently proposed stick-breaking scheme for the output layer [27] but also tested with the standard softmax function. As shown in the following sections, combining these two elements results in improved performance for ordinal problems with respect to previously published alternatives.

The rest of this paper is organised as follows: previous related works, including the stick-breaking and the loss regularisation, are described in Section 2; in Section 3, the new cross-entropy loss regularisation with the beta distribution is explained; in Section 4, the design of the experiments and the datasets used are described; in Section 5, the results of the experiments are shown and compared with previous works; and, finally, in Section 6, the conclusions of this work are presented.

2. Related works

2.1. Stick-breaking

The stick-breaking approach considers the problem of breaking a stick of length 1 into J segments. This methodology is related to non-parametric Bayesian methods and can be considered a subset of the random allocation processes [28]. Also, this method has been applied as a generalisation of the continuation ratio models [29].

In Latent Gaussian Models (LGMs), the latent variables follow a Gaussian distribution where the probability of each categorical variable is parametrised in terms of a linear projection $\eta_1, \ldots, \eta_J$ [30], where J is the number of classes of the problem. The probability of the first category is modelled as $\sigma(\eta_1)$ where $\sigma(x) = 1/(1+e^{-x})$. This represents the first piece of the stick. The length of the remainder of the stick is $(1-\sigma(\eta_1))$. The probability of the second category is a fraction $\sigma(\eta_2)$ of the remainder of the stick. Following this process, we can define the probability of each of the J categories. The probabilities (stick lengths) are all positive and sum to one; they thus define a valid probability distribution. A different function for $\sigma(x)$ can be used (such as the probit function). However, the logit allows us to use efficient variational bounds. The stick-breaking parametrisation can be written compactly as:

$$p(y=C_j|\eta_q) = \prod_{i=1}^{j-1}(1-\sigma(\eta_i))\sigma(\eta_j),\ j=1,\ldots,J. \tag{1}$$

Recently, a stick-breaking approach was presented in [27] as an alternative to the standard softmax for ordinal classification problems where the output distribution should be unimodal. The authors define a stick of unit length and sequentially break off parts of the stick. The length of the generated stick fragments can represent the discrete probabilities for each class.

In the first stick-breaking, two parts with the length of $\sigma(\eta_1)$ and $1-\sigma(\eta_1)$ are created. These fragments represent the probability of the first class:

$$p(y_i = C_1) = \sigma(\eta_1) = \sigma(f_1(x)) = \frac{1}{1+e^{-f_1(x)}}, \tag{2}$$

and the probability of the remaining classes in the ordinal scale $(y_i \succ C_1)$:

$$\begin{aligned} p(y_i \succ C_1) &= 1-\sigma(\eta_1) = 1-\sigma(f_1(x)) = \\ &= 1 - \frac{1}{1+e^{-f_1(x)}} = \frac{1}{1+e^{f_1(x)}}. \end{aligned} \tag{3}$$

Then, the remaining part $1-\sigma(\eta_1)$ is broken, obtaining two parts of length $\sigma(\eta_2)(1-\sigma(\eta_1))$ and $(1-\sigma(\eta_2))(1-\sigma(\eta_1))$.

This breaking process can be mathematically written as:

$$p(y=C_1|\eta_1) = l_1 = \sigma(\eta_1), \tag{4}$$

$$\begin{aligned} p(y=C_j|\{\eta_k\}_{k=1}^{j}) &= l_j = \\ &= \sigma(\eta_j)\prod_{i=1}^{j-1}(1-\sigma(\eta_i)), \quad j=2,\ldots,J-1, \end{aligned} \tag{5}$$

$$\begin{aligned} p(y=C_J|\{\eta_k\}_{k=1}^{J-1}) = l_J &= \prod_{i=1}^{J-1}(1-\sigma(\eta_i)) = \\ &= \prod_{i=1}^{J-1}\frac{1}{1+e^{f_i(x)}}, \end{aligned} \tag{6}$$

where the length of each bit can be used to formulate the probability of each class.

The stick-breaking process is used for training deep ordinal neural networks [27]. To do this, the authors set $J-1$ output neurons for a problem with J ranks or ordinal categories. $f_i(x)$ is a scalar denoting the ith output of the neural network and replaces the linear projections (η_i) of the LGMs. The conventional cross-entropy loss, CE, can be used to train the model.

It can be derived that each output associated with $f_i(x)$ is actually the ratio:

$$\begin{aligned} \frac{p(y=C_i|x)}{p(y \succcurlyeq C_i|x)} &= \frac{p(y=C_i|x)}{p(y=C_i|x)+p(y\succ C_i|x)} = \\ &= \frac{\frac{e^{f_i(x)}}{1+e^{f_i(x)}}\prod_{l=1}^{i-1}\frac{1}{1+e^{f_l(x)}}}{\frac{e^{f_i(x)}}{1+e^{f_i(x)}}\prod_{l=1}^{i-1}\frac{1}{1+e^{f_l(x)}}+\prod_{l=1}^{i}\frac{1}{1+e^{f_l(x)}}} = \\ &= \frac{e^{f_i(x)}}{1+e^{f_i(x)}} = \frac{1}{1+e^{-f_i(x)}} = \sigma(f_i(x)). \end{aligned} \tag{7}$$

Consequently, $f_i(x)$ can be interpreted as defining decision boundaries that try to separate the ith class from all the classes that come after it. By doing so, the prediction is still a discrete probability.

An interesting property of this method is that, unlike other approaches that only output a single distribution value [25,31], it is more expressive, because each boundary of two adjacent classes has its own scalar output $f_i(x)$.

2.2. Unimodal regularisation

Label smoothing is a general regularisation to address the noisy label problem, which encourages the model to be less confident [27]. In the case of a one-hot label, the distribution of a label probability is $q(i) = \delta_{i,1}$, where 1 is the ground truth class, $\delta_{i,1}$ is a Dirac delta, which equals to 1 for $i = 1$, and 0 otherwise.

This label smoothing can be applied to the cross-entropy loss and replaces $q(i)$:

$$L = \sum_{i=1}^{J} q(i)[-\log p(y = C_i|x)] \tag{8}$$

with a more conservative target distribution:

$$L = \sum_{i=1}^{J} q'(i)[-\log p(y = C_i|x)] \tag{9}$$

where $q'(i) = (1-\eta)\delta_{i,1} + \eta\frac{1}{J}$ and η is the parameter that controls the linear combination.

In ordinal classification, errors in classifying a pattern in its real class are more likely to be caused by the classifier classifying them in the closest classes. Therefore, building unimodal distributions which have their mode in the centre of the interval, for the case of middle classes, or in the upper or lower bounds, for extreme classes, should report a more accurate loss computation. Moreover, it is quite important that the probability distribution has small variance and the majority of its probability mass is concentrated in the interval associated with the real class. In this way, the probability is drastically reduced as long as we go further from the correct class.

The distributions proposed in previous works [25–27] to model the targets in a soft manner have improved the performance of ordinal classifiers concerning the standard one-hot encoding. However, they have high variance or do not offer the required flexibility to position the mode of each distribution in the centre of the class interval while preserving a small variance. Also, some of the proposed methods require to adjust experimentally different parameters.

In [25], the authors used Poisson distributions to model the probabilities. The mean and variance of this kind of distributions is equal to the distribution parameter λ. Therefore, it has limited flexibility to obtain a small variance. For this reason, they also used the binomial distribution, which has two parameters: the number of classes, J, and the probability, p. Even though the mean (Jp) and the variance ($Jp(1-p)$) have different expressions, it is not easy to position the mode in the right point of the interval while obtaining a small variance. Finally, the authors of [27] proposed to sample on an exponential function $e^{\frac{-|i-l|}{\tau}}$, where l is the class of the pattern and $i = 1, \dots, J$, followed by a softmax normalisation. However, the value of τ must be adjusted experimentally and, in some cases, the probability mass is not sufficiently concentrated in the interval of the correct class.

To overcome the issues related to the Poisson and binomial distributions and the exponential function described, we propose in this work a set of probability distributions associated with the beta distribution, given that their variance is small and the domain of the distribution is between 0 and 1. As a graphical example, Fig. 1 illustrates the shape of the distribution associated with each class and type of distribution for a problem with five classes. For the discrete distributions, the class number is represented in the x axis while the class intervals are used for the continuous distributions.

3. Proposed method

3.1. Beta regularised cross-entropy loss

The main idea behind this work is to use probability distributions to model the targets as unimodal distributions instead of using the one-hot encoding. In this way, we obtain the soft target distributions $q'(i)$ discussed in Section 2. To do that, we consider the beta distribution defined in the range [0,1], therefore there is no need to apply any normalisation, and, also, it does not lead to high variance. The beta distribution has been applied to model the behaviour of random variables limited to intervals of finite lengths in a wide variety of disciplines. Some properties of this distribution are described below.

In its standard form, the beta distribution, $\beta(a, b)$, is a continuous distribution, and its probability density function (pdf) is:

$$f(x, a, b) = \frac{x^{a-1}(1-x)^{b-1}}{B(a, b)}, \tag{10}$$

where $0 < x < 1$, $a > 0$ and $b > 0$. This is also known as the classical beta distribution or the Incomplete Beta function. The function $B(a, b)$ has the form:

$$B(a, b) = \int_0^1 x^{a-1}(1-x)^{b-1}dx = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}, \tag{11}$$

where $\Gamma(a) = (a-1)!$. When $a, b > 1$, $f(x)$ has a unique mode at $\frac{a-1}{(a+b-2)}$ and is zero at $x = 0$ and $x = 1$. If $a = 1$ or $b = 1$, then $f(x)$ has a corresponding terminal value b or a. Finally, if $a = b = 1$, then $f(x)$ becomes the uniform distribution.

Since the range of $f(x)$ is finite, all its moments exist. Its mean is given by the expression $E(x) = \frac{a}{a+b}$ and its variance is defined as $V(x) = \frac{ab}{(a+b)^2(a+b+1)}$.

In order to analyse the behaviour of this distribution, we consider an ordinal classification problem with five classes ($J = 5$). We assume that the distributions of the labels are beta distributions, class 1 takes random values in the range [0,0.2], class 2 in the range [0.2,0.4], class 3 in [0.4,0.6], class 4 in [0.6,0.8], and class 5 in [0.8,1.0].

First, we find the value for the parameters a and b which makes the beta distribution associated to the class 1 have its centre in the middle of the [0,0.2] interval. In this case, we chose $a = 1$ and $b = 9$, what leads to $E(x) = 0.1$ and $V(x) = 0.008$. The associated density function is given by:

$$f(x) = 9(1-x)^8, \quad 0 \leq x \leq 1. \tag{12}$$

In this way, the probability of each class can be calculated as follows:

$$p_1 = p(y = C_1) = \int_0^{0.2} 9(1-x)^8 dx = 0.8758, \tag{13}$$

and, therefore, $p_2 = 0.1241$, $p_3 = 0.0098$, $p_4 = 2.6 \times 10^{-4}$ and $p_5 = 5.1 \times 10^{-6}$.

In the same way, we can compute the distributions and the probabilities associated with the other classes finding the a and b parameters that make the distribution be centred in the intervals [0.2,0.4], [0.4,0.6], [0.6,0.8] and [0.8,1.0].

However, adjusting these parameters by trial and error is not the best method, as it requires additional computational time. So, in the next section, an alternative method to determine a and b is proposed.

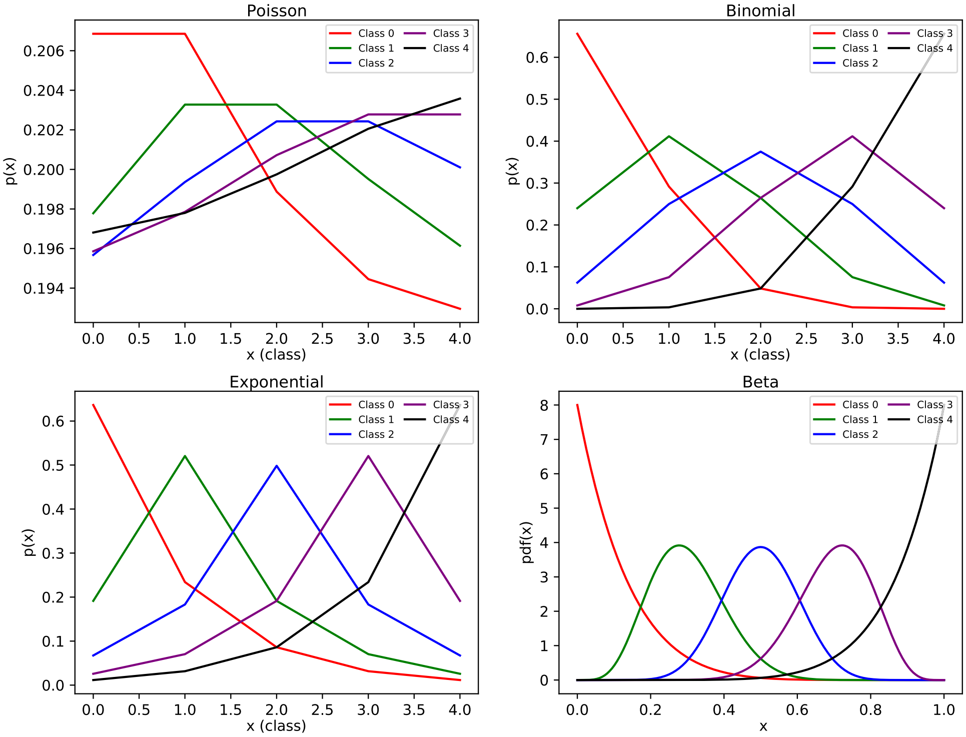


Fig. 1. Label distributions shape for $N = 5$ and different real classes: 0 (red), 1 (green), 2 (blue), 3 (purple), 4 (black). The x axis represents the labels for the discrete distributions and the class intervals for the beta. The y axis shows the probability for the Poisson, binomial and exponential distributions and the value of the probability density function for the beta distribution. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. Beta distribution parameters based on number of classes

In this section, we propose a method to find the parameters (a, b) for each class on any ordinal problem based on the number of classes.

First, we define the thresholds of each class based on the number of classes (J). For any value of J, the centres of the intervals can be obtained as $1/(2J)$, $3/(2J)$, ..., $(2J-1)/(2J)$. The length of the first interval should be $1/J$, and the mean value is $1/(2J)$. Consequently:

$$E(x) = \frac{a}{a+b} = \frac{1}{2J}, \Rightarrow b = a(2J-1). \tag{14}$$

Then, the variance can be defined as:

$$V(x) = \frac{2J-1}{4J^2(2Ja+1)}, \tag{15}$$

and the standard deviation:

$$S(x) = \frac{1}{2J}\sqrt{\frac{2J-1}{2Ja+1}}. \tag{16}$$

We assume that most of the values of the distribution should be in the range $E(x) \pm S(x)$. In this way, we obtain the constraints for the first interval:

$$0 < \frac{1}{2J} \pm \frac{1}{2J}\sqrt{\frac{2J-1}{2Ja+1}} < \frac{1}{J}. \tag{17}$$

Solving the first inequality, we get:

$$\frac{2J-1}{2Ja+1} < 1 \Rightarrow a > \frac{J-1}{J}. \tag{18}$$

As a consequence, for any J, we can use $a = 1$ and $b = a(2J-1)$.

In the same way, we obtain the parameters for the second interval. Now, $E(x) = \frac{a}{a+b} = \frac{3}{2J}$ and $b = \frac{a(2J-3)}{3}$. The variance and the standard deviation are:

$$V(x) = \frac{9(2J-3)}{4J^2(2Ja+3)}, \tag{19}$$

$$S(x) = \frac{3}{2J}\sqrt{\frac{2J-3}{2Ja+3}}. \tag{20}$$

And the constraints for this interval are given by:

$$\frac{1}{J} < \frac{3}{2J} \pm \frac{3}{2J}\sqrt{\frac{2J-3}{2Ja+3}} < \frac{2}{J}, \tag{21}$$

what leads to:

$$\frac{2J-3}{2Ja+3} < \frac{1}{9} \Rightarrow a > \frac{9(2J-3)-3}{2J}. \tag{22}$$

In the same way, we can obtain the parameters for the rest of the intervals. The parameters of the beta distributions for each class are shown in Table 1 for different number of classes ($J \leq 8$).

Finally, the beta regularised cross-entropy loss can be expressed as:

$$L = \sum_{i=1}^{J} q'(i)[-\log p(y = C_i|x)], \tag{23}$$

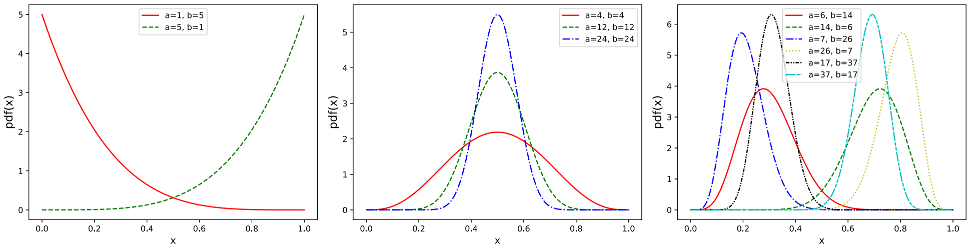
where $q'(i) = (1-\eta)\delta_{i,1} + \eta f(x, a, b)$ and $f(x, a, b)$ is the probability value sampled from a beta distribution that is centred in $x = \frac{2J-1}{2J}$ and uses the a and b parameters obtained using the method described in this section.

3.3. Beta distribution properties regarding ordinal classes representation

The main benefit of using the beta distribution for modelling the probability distribution is the fact that it has two parameters that allow obtaining different distribution shapes with small

Table 1
Beta parameters (a,b) for each class and each number of classes.

	(a, b) parameters for class							
J	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
3	(1,4)	(4,4)	(4,1)					
4	(1,6)	(6,10)	(10,6)	(6,1)				
5	(1,8)	(6,14)	(12,12)	(14,6)	(8,1)			
6	(1,10)	(7,20)	(15,20)	(20,15)	(20,7)	(10,1)		
7	(1,12)	(7,26)	(16,28)	(24,24)	(28,16)	(26,7)	(12,1)	
8	(1,14)	(7,31)	(17,37)	(27,35)	(35,27)	(37,17)	(31,7)	(14,1)



(a) $a = 1$ or $b = 1$.

(b) $a = b$.

(c) $a \neq 1$, $b \neq 1$ and $a \neq b$

Fig. 2. Beta distribution shapes for extreme, middle and intermediate classes.

variance. Thus, it can represent the distribution of extreme classes along with the symmetric distribution of the middle class. When $a = 1$, the probability density function is given by:

$$f(x, 1, b) = b(1-x)^{b-1}, \quad 0 \le x \le 1, \tag{24}$$

while the pdf for $b = 1$ is:

$$f(x, a, 1) = ax^{a-1}, \quad 0 \le x \le 1. \tag{25}$$

As shown in Fig. 2a, these functions can easily represent the distributions associated with the extreme classes.

On the other hand, the beta distribution with $a = b$ and $b \to \infty$ is similar to a normal distribution. Let the random variable X be associated to a $\beta(b, b)$ distribution with probability density function:

$$f_X(x, b) = \frac{\Gamma(2b)(x(1-x))^{b-1}}{\Gamma^2(b)}, \quad 0 \le x \le 1, \tag{26}$$

where b is a real positive parameter. The mean of X is $E[X] = 0.5$ and the variance of X is $V[X] = \frac{1}{4(2b+1)}$.

Proposition The $\beta(b, b)$ distribution converges to the normal distribution when $b \to \infty$, that is:

$$\beta(b, b) \xrightarrow{d} N\left(\frac{1}{2}, \frac{1}{4(2b+1)}\right). \tag{27}$$

The proof of this proposition is included in Appendix A.

Therefore, the beta distribution can also accurately represent the distribution of the middle class through a symmetric distribution when the problem has an odd number of classes keeping the variance small (see Fig. 2b). When the value of b increases, the variance of the distribution becomes smaller. The symmetric property of the distribution in the aforementioned cases can be easily checked with the skewness coefficient, which is calculated as:

$$\text{Skewness} = \frac{2(b-a)\sqrt{a+b+1}}{(a+b+2)\sqrt{ab}}, \tag{28}$$

which is zero due to the fact that $a = b$.

As mentioned before, the rest of the classes have asymmetrical distributions with small variance, as can be observed in Fig. 2c. However, when the number of classes increases, the resulting distributions gets closer to a normal distribution with small variance (see distribution for (17,37) or (37,17) in Fig. 2c).

The properties described in this section make the beta distribution be an excellent choice for modelling the probability distribution of each class in an ordinal problem, as it can precisely represent both the extreme and the middle classes.

4. Experiments

4.1. Data

The ordinal classification of images has not been widely explored yet and, therefore there are not many ordinal images benchmark datasets that can be used to test our approach. We have evaluated the different proposals using the most well-known ordinal images datasets.

4.1.1. Diabetic Retinopathy

Diabetic Retinopathy (DR) is a dataset consisting of extremely high-resolution fundus image data. It was used in a Kaggle competition[1] and has been used in several previous works [32,33] as a benchmark dataset for ordinal classification. The training set consists of 17563 pairs of images (where a pair includes a left and right eye image corresponding to a patient). In this dataset, we try to predict the correct category from five levels of DR: no DR (25810 images), mild DR (2443 images), moderate DR (5292 images), severe DR (873 images), or proliferative DR (708 images). The test set contains 26788 pairs of images, which are distributed in the same five classes with the following proportions: 39532, 3762, 7860, 1214 and 1208 images. These images are taken in variable conditions: by different cameras, conditions of illumination and resolutions. They come from the EyePACS dataset that was used in the DR detection competition hosted on the Kaggle platform. The images are resized to 128 × 128 pixels and the value of each pixel is standardised using the mean and the standard deviation of the training set. Some images from the test set are presented in Fig. 3a.

4.1.2. Adience

Adience[2] dataset consists of 26580 faces belonging to 2284 subjects. It has been used in previous works [34] for gender and

[1] https://www.kaggle.com/c/diabetic-retinopathy-detection/data

[2] http://www.openu.ac.il/home/hassner/Adience/data.html

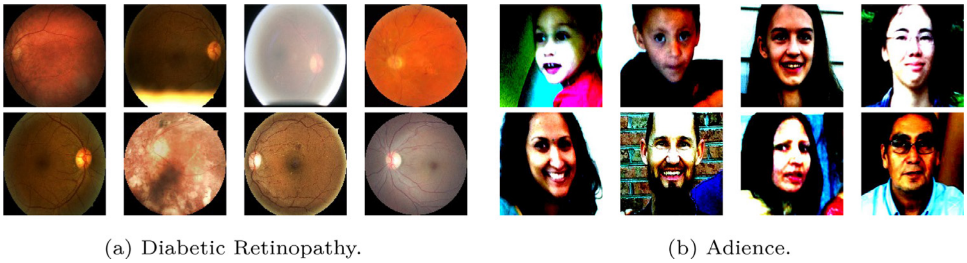


Fig. 3. Examples from different classes taken from the test set of Diabetic Retinopathy and Adience.

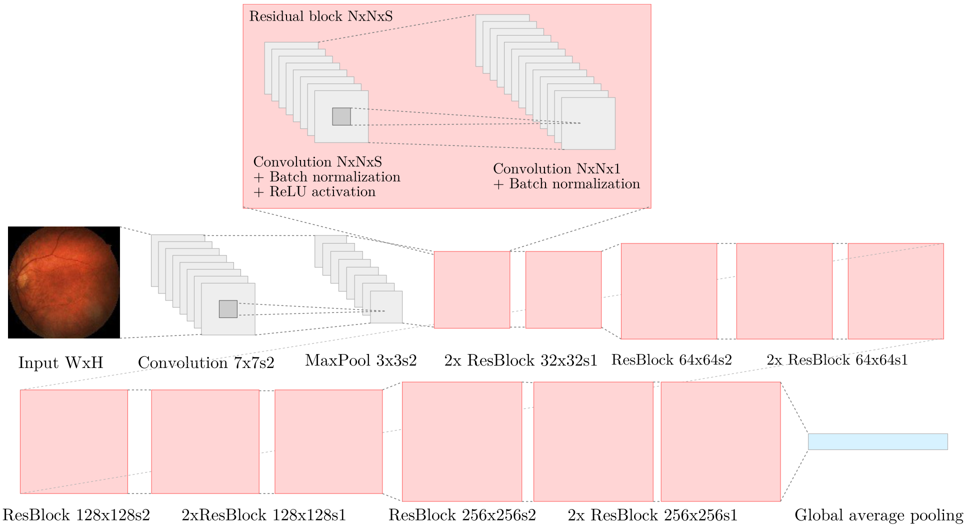


Fig. 4. Network architecture.

age classification. The faces that appear in the original images of this dataset have been pre-cropped and aligned in order to ease the training process. Also, images have been resized to 256 × 256 pixels and contrast-normalised and the distribution of the pixels was standardised. The original dataset was split into five cross-validation folds. The training set consists of merging the first four folds which comprise a total of 15554 images. The last fold is used as test set. Fig. 3b shows some images taken from the test set.

4.1.3. FGNet

FGNet[3] is the smallest dataset considered in this work. It consists of 1002 128 × 128 colour images of faces from 82 different subjects. From these images, we took 80% for training and the remaining 20% for testing. These partitions were done in a stratified way. Each image was labelled with the exact age that the subject had at the moment that the picture was taken. We grouped these ages into six categories based on age ranges (0-3, 3-11, 11-16, 16-24, 24-40, $>$ 40).

4.2. Model

The model considered for this work is a Residual Convolutional Network [27], as it can achieve good generalisation capabilities with a reduced number of parameters. Figure 4 shows more details about the layers that compose the network architecture. Kernel size and stride is specified for each convolutional and pooling layer. The structure of a residual block ResBlock NxNsS is shown in Fig. 4 too. The output of each residual block is concatenated with the input. The parameters of every convolutional and batch normalisation layer are L2 normalised (10^{-4}). He normal initialisation [35] has been used for the weights and bias of these layers. The global average pooling layer replaces each channel of its input with the mean value of all the pixels of the channel. This layer achieves a high reduction of data dimensionality, significantly reducing the number of parameters at the end of the network while obtaining good performance.

In the output layer of the model, two different alternatives are considered: (1) a dense layer with N units and the standard softmax function, (2) a dense layer with $N-1$ neurons, sigmoid activation and followed by the stick breaking layer described in Section 2.1.

4.3. Experimental design

The model described above is trained using the three datasets described in Section 4.1. The convolutional network model was optimised using the well-known batch based first-order optimisation algorithm called Adam [36]. The initial learning rate ($\eta = 10^{-4}$) of the optimiser and the batch size (128) were adjusted by cross-validation. The training process is run for 100 epochs and repeated 10 times following a 10-fold validation scheme, where we take 9 folds for training and the remaining for validation. To ease further comparison and make possible the reproducibility of the experiments, the folds considered were the same for all the experiments.

Using the aforementioned validation set, an early stopping mechanism is applied in order to stop the training process when the validation loss has not improved for 20 epochs. Also, when the validation loss has not decreased for 8 epochs, the learning rate will be multiplied by a 0.5 factor until it reaches 10^{-6}.

Data augmentation techniques are applied as previous works [37] have proved that they avoid the model over-fitting and

[3] https://yanweifu.github.io/FG_NET_data/index.html

Table 2
Results for each dataset and method.

Retinopathy					
Method	QWK	CCR	1-off	MS	Time (s)
S CE	$0.0839_{0.0182}$	$\mathbf{0.7253}_{\mathbf{0.0046}}$	$\mathit{0.7987}_{\mathit{0.0043}}$	$0.0012_{0.0015}$	$2329.31_{107.87}$
S CE-P	$0.1018_{0.0161}$	$0.6285_{0.0597}$	$0.7398_{0.0514}$	$0.0099_{0.0166}$	$2415.96_{36.91}$
S CE-B	$\mathit{0.1249}_{\mathit{0.0199}}$	$0.6653_{0.0368}$	$0.7959_{0.0204}$	$\mathit{0.0121}_{\mathit{0.0159}}$	$2402.81_{67.65}$
S CE-E	$0.1082_{0.0182}$	$0.6833_{0.0458}$	$0.7904_{0.0129}$	$0.0108_{0.0056}$	$2371.53_{203.26}$
S CE-β	$0.1115_{0.0152}$	$0.7097_{0.0095}$	$0.7846_{0.0082}$	$0.0002_{0.0004}$	$2140.22_{68.35}$
SB CE	$0.0941_{0.0179}$	$\mathit{0.7145}_{\mathit{0.0121}}$	$0.7914_{0.0086}$	$0.0046_{0.0034}$	$\mathbf{1394.49}_{\mathbf{119.91}}$
SB CE-P	$0.0937_{0.0197}$	$0.6673_{0.0534}$	$0.7757_{0.0338}$	$\mathbf{0.0126}_{\mathbf{0.0090}}$	$2421.71_{41.02}$
SB CE-B	$\mathbf{0.1285}_{\mathbf{0.0082}}$	$0.6762_{0.0133}$	$0.7962_{0.0032}$	$0.0067_{0.0033}$	$2414.11_{47.94}$
SB CE-E	$0.0989_{0.0202}$	$0.6745_{0.0554}$	$\mathbf{0.7993}_{\mathbf{0.0168}}$	$0.0086_{0.0057}$	$2425.29_{46.06}$
SB CE-β	$0.1093_{0.0198}$	$0.7122_{0.0120}$	$0.7876_{0.0104}$	$0.0002_{0.0005}$	$\mathit{2102.78}_{\mathit{108.06}}$
			Adience		
Method	QWK	CCR	1-off	MS	Time (s)
S CE	$0.6604_{0.0249}$	$0.3903_{0.0267}$	$0.7132_{0.0158}$	$0.0296_{0.0204}$	$\mathbf{972.12}_{\mathbf{104.77}}$
S CE-P	$0.6558_{0.0370}$	$0.3536_{0.0480}$	$0.7095_{0.0435}$	$0.0172_{0.0269}$	$3416.035_{300.92}$
S CE-B	$\mathit{0.7403}_{\mathit{0.228}}$	$0.4063_{0.0290}$	$\mathbf{0.7795}_{\mathbf{0.0175}}$	$0.0296_{0.0191}$	$3637.84_{19.22}$
S CE-E	$0.7279_{0.0213}$	$0.3997_{0.0442}$	$\mathit{0.7691}_{\mathit{0.0218}}$	$0.0382_{0.0287}$	$3644.54_{24.84}$
S CE-β	$0.7306_{0.0136}$	$\mathbf{0.4160}_{\mathbf{0.0128}}$	$0.7647_{0.0135}$	$0.0938_{0.0481}$	$2911.61_{474.94}$
SB CE	$0.7016_{0.0162}$	$0.3975_{0.0128}$	$0.7477_{0.0072}$	$\mathit{0.0946}_{\mathit{0.0219}}$	$2717.54_{271.43}$
SB CE-P	$0.5794_{0.0881}$	$0.2954_{0.0705}$	$0.6884_{0.0498}$	$0.0172_{0.0233}$	$3789.46_{453.72}$
SB CE-B	$0.7133_{0.0598}$	$0.3870_{0.0514}$	$0.7662_{0.0305}$	$0.0240_{0.0352}$	$3386.15_{14.52}$
SB CE-E	$0.7246_{0.0438}$	$0.3915_{0.0622}$	$0.7671_{0.0246}$	$0.0490_{0.0295}$	$3636.84_{19.71}$
SB CE-β	$\mathbf{0.7416}_{\mathbf{0.0078}}$	$\mathit{0.4123}_{\mathit{0.0102}}$	$0.7640_{0.0082}$	$\mathbf{0.0955}_{\mathbf{0.0252}}$	$\mathit{2643.58}_{\mathit{557.93}}$
			FG-Net		
Method	QWK	CCR	1-off	MS	Time (s)
S CE	$0.4855_{0.0689}$	$0.3844_{0.0318}$	$0.7449_{0.0282}$	$0.1275_{0.0664}$	$\mathit{92.27}_{\mathit{35.16}}$
S CE-P	$0.4621_{0.0505}$	$0.3355_{0.0318}$	$0.7206_{0.0357}$	$0.1267_{0.0801}$	$121.63_{15.54}$
S CE-B	$\mathit{0.6452}_{\mathit{0.0378}}$	$\mathit{0.3899}_{\mathit{0.0267}}$	$\mathit{0.8182}_{\mathit{0.0319}}$	$0.1500_{0.0597}$	$108.90_{17.68}$
S CE-E	$0.6118_{0.0375}$	$0.3894_{0.0262}$	$0.7959_{0.0203}$	$\mathit{0.1764}_{\mathit{0.0678}}$	$111.27_{20.58}$
S CE-β	$0.6037_{0.0551}$	$\mathbf{0.3934}_{\mathbf{0.0302}}$	$0.7959_{0.0258}$	$\mathbf{0.2071}_{\mathbf{0.0642}}$	$105.27_{24.41}$
SB CE	$0.5478_{0.1650}$	$0.3768_{0.0578}$	$0.7529_{0.1035}$	$0.1367_{0.0652}$	$\mathbf{86.46}_{\mathbf{16.16}}$
SB CE-P	$0.4907_{0.0677}$	$0.3381_{0.0248}$	$0.7342_{0.0279}$	$0.1153_{0.0621}$	$121.18_{19.14}$
SB CE-B	$\mathbf{0.6594}_{\mathbf{0.0394}}$	$0.3634_{0.0257}$	$\mathbf{0.8212}_{\mathbf{0.0256}}$	$0.1407_{0.0601}$	$104.38_{17.48}$
SB CE-E	$0.6293_{0.0467}$	$0.3723_{0.0237}$	$0.8048_{0.0259}$	$0.1524_{0.0520}$	$106.77_{19.26}$
SB CE-β	$0.6416_{0.0334}$	$0.3791_{0.0275}$	$0.8027_{0.0267}$	$0.1744_{0.0468}$	$97.88_{17.60}$

reduce the amount of data needed to train a deep learning model. We considered the following transformations: horizontal flipping, random zoom in or out within a $[-20\%, 20\%]$ range and random width shifting within a $[-10\%, 10\%]$ range. They are individually applied to every image in the training set with a certain probability. In this way, more than one transformation can be applied to the same image. Also, for the zoom in/out and the width shifting, the magnitude of the transformation is randomly selected from the ranges described.

In terms of the loss function used for the optimisation algorithm, we have considered five different alternatives, all based on the standard cross-entropy loss:

- Standard cross-entropy.
- Cross-entropy loss with poisson regularisation (CE-P) [26].
- Cross-entropy loss with binomial regularisation (CE-B) [26].
- Cross-entropy loss with exponential regularisation (CE-E) [27].
- Cross-entropy loss with the beta regularisation (CE-β) proposed in this work (Section 3.1). The parameters used for the distribution are obtained using the method described in Section 3.2.

Since datasets are imbalanced, the loss function is weighted based on the a priori probabilities of the classes (considering the number of instances of each class in the training set) following the method described in [38]. Classes with few samples have a higher weight than classes with many instances.

Considering the different alternatives for the output layer described in Section 4.2 and the separate loss functions described in this Section, ten different experiments were run. As mentioned before, each of these experiments was repeated ten times using the described 10-fold cross-validation scheme. These experiments can be reproduced running the code available in our public repository[4].

5. Results

The results of the experiments described in Section 4 are presented in this section. The evaluation metrics used are the Quadratic Weighted Kappa (QWK) [33], the Correct Classification Rate (CCR) or accuracy, the Minimum Sensitivity (MS) [39] and the execution time. All the values presented in Table 2 are the mean and the standard deviation of all the executions ran for each method in the test set. The experiments with softmax in the output layer are denoted as S and the experiments using the stick-breaking scheme as SB. The best result of each metric is highlighted in bold font face, while the second one is in italics. All the metrics must be maximised, except the execution time.

5.1. Statistical analysis

In this Section, a statistical analysis have been carried out in order to obtain robust conclusions from the experimental results. Each of the metrics presented in Section 5 were analysed separately.

First, the Kolmogorov-Smirnov test for the QWK reported that the values of this metric are normally distributed. Then, an ANOVA

[4] https://github.com/victormvy/beta-regularisation-cnn

Table 3
HSD Tukey's test results for QWK (30 samples for each method).

Method	Subsets					
	1	2	3	4	5	6
S CE-P	0.4288					
SB CE-P	0.4291					
SB CE		0.5062				
S CE		0.5111	0.5111			
S CE-β		0.5307	0.5307	0.5307		
S CE-E			0.5343	0.5343	0.5343	
S CE-E				0.5423	0.5423	0.5423
SB CE-β				0.5551	0.5551	0.5551
S CE-B					0.5602	0.5602
SB CE-B						0.5640
	p-values	0.1000	0.1450	0.1000	0.0640	0.0000

Table 4
HSD Tukey's test results for CCR (30 samples for each method).

Method	Subsets			
	1	2	3	4
SB CE-P	0.3954			
S CE-P	0.3977			
SB CE-B		0.4307		
SB CE-E		0.4366	0.4366	
S CE-B		0.4483	0.4483	0.4483
S CE-E		0.4502	0.4502	0.4502
SB CE		0.4510	0.4510	0.4510
SB CE-β			0.4523	0.4523
S CE			0.4538	0.4538
S CE-β				0.4612
	p-values	0.0640	0.2130	0.6240

II Test [40] with the method and the dataset as factors was performed, in order to check whether these factors had any impact on the value of these metric. The parametric test reported that both factors were significant (p-value < 0.001) and that there is an interaction between them.

Given that the factors considered are significant, a posthoc HSD Tukey's test was performed [41]. The results of this test are shown in Table 3. The stick breaking with cross-entropy binomial regularised loss obtained the best mean results. However, there are no significant differences with S CE-B, SB CE-β and SB CE-E.

The same analysis was carried out for the CCR metric. The Kolmogorov-Smirnov reported that the values are normally distributed, and the ANOVA II test found significant influence of the factors considered as well as an interaction between them. The posthoc test results are shown in Table 4. In this case, the best methodology is the one that uses the softmax output layer combined with the beta regularisation for the cross-entropy loss. However, there are no significant differences with S, S CE-β, SB, S CE-E and S CE-B.

In the case of the MS metric, the values are also normally distributed, and there are significant differences based on the two factors considered. The posthoc Tukey's test results are displayed in Table 5. Again, the best method was the one that uses softmax in the output layer and the beta regularised cross-entropy loss. However, there are no significant differences with SB CE-β and S CE-E.

Finally, the experiment time is also normally distributed, and the ANOVA II test reported significant differences based on the factors considered. The posthoc test (Table 6) showed that the method with the best average time is the standard softmax coupled with the cross-entropy, followed by the stick-breaking with cross-entropy. Within the methods that use regularisation, the beta regularised cross-entropy with softmax or stick-breaking is the one with the best time.

Table 5
HSD Tukey's test results for MS (30 samples for each method).

Method	Subsets			
	1	2	3	4
SB CE-P	0.0752			
S CE-P	0.0814			
S CE	0.0827			
SB CE-B	0.0906	0.0906		
S CE-B	0.0983	0.0983		
S CE-E	0.1029	0.1029	0.1029	
SB CE	0.1049	0.1049	0.1049	
S CE-E		0.1156	0.1156	0.1156
SB CE-β			0.1238	0.1238
S CE-β				0.1430
	p-values	0.2670	0.2480	0.1590

Table 6
HSD Tukey's test results for Time (30 samples for each method).

Method	Subsets				
	1	2	3	4	5
S CE	715.65				
SB CE		875.62			
SB CE-β			1007.00		
S CE-β			1073.53		
SB CE-B				1222.68	
S CE-P				1239.38	1239.38
S CE-E				1269.98	1269.98
S CE-B				1273.47	1273.47
SB CE-E				1276.49	1276.49
SB CE-P					1314.94
	p-values	1.0000	0.3700	0.6580	0.1810

When we analyse the results of all the metrics combined, we find that the method that uses stick breaking with beta regularised loss (SB CE-β) achieves the best result for QWK and CCR, and the second best for MS. Also, as mentioned before, it obtains the best time among the methods that use regularisation. These facts turn this method into a competitive alternative that can be applied to solve other ordinal classification problems.

6. Conclusions

In this work, we have proposed the application of a unimodal regularisation based on beta distributions for the cross-entropy loss. The method described improved the performance on problems where classes follow a natural ordering. The regularisation proposed benefit from the fact that, in ordinal regression problems, misclassification tends to be in adjacent classes, and, consequently, slightly modifying the labels considering the ordinal scale should increase the robustness of the model in the presence of noisy targets. Thus, the main advantage of the proposed regularised loss is that it encourages the classification errors to be in the adjacent classes and minimises the number of errors in distant classes, achieving more accurate results for ordinal problems.

The distribution used to regularise the loss function has two parameters. Therefore, a method to automatically determine these parameters has been introduced. This method avoids learning them from the training data, thus improving the computational time with respect to other alternatives with free parameters to be adjusted. The parameters obtained through this method have been used for the label smoothing that has been applied as a regularisation method for the loss function and tested with three datasets and one CNN model. Even though the model used was a deep learning method, the proposal of this work is also suitable for other kinds of modelling techniques.

This regularised loss has been combined, in one hand, with the standard softmax function in the output layer and, in the other, with the stick-breaking method. Moreover, it has been also compared with previously proposed alternatives as well as the standard nominal classification methods. The statistical tests that were carried out with the obtained results showed that the proposed method improves the performance on ordinal problems for several metrics. Also, these tests corroborated an interaction between the ordinal method and the dataset considered, which means that some methodologies are more accurate than others for some datasets. However, the stick-breaking with beta regularised cross-entropy achieved the best global results when analysing the three datasets. Therefore, the proposed method has significantly improved the performance on the benchmark ordinal problems and, in the future, can be applied to real world problems that have an underlying ordinal structure.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported in part by the Agencia Espaola de Investigación under Grant PID2020-115454GB-C22, in part by the Consejería de Economía, Conocimiento, Empresas y Universidad de la Junta de Andalucía under Grant UCO-1261651, in part by the Consejería de Transformacin Económica, Industria, Conocimiento y Universidades (Junta de Andalucía) y Programa Operativo FEDER 2014-2020 under Grant PY20_00074, and in part by the Consejera de Salud y Familia de la Junta de Andaluca under Grant PS-2020-780. The work of Víctor Manuel Vargas was supported in part by the FPU Predoctoral Program of the Spanish Ministry of Science, Innovation and Universities (MCIU) under Grant FPU18/00358. The open access charge was fully supported by Universidad de Córdoba / CBUA.

Appendix A. Beta convergence to a normal distribution

If we subtract the mean and divide by the standard deviation and we make a change of scale and origin, the proposition can be written in a more particular form:

$$Y = g(X) = 2\sqrt{(2b+1)}\left(X - \frac{1}{2}\right) \to N(0,1),$$

where $-\sqrt{2b+1} < y < \sqrt{2b+1}$.

$$\begin{aligned} G_Y(y) &= P(Y \le y) = P\left(2\sqrt{(2b+1)}\left(X - \frac{1}{2}\right) \le y\right) = \\ &= P\left(X \le \frac{Y}{2\sqrt{2b+1}} + \frac{1}{2}\right) = F\left(\frac{Y}{2\sqrt{2b+1}} + \frac{1}{2}\right). \end{aligned}$$

$$\begin{aligned} g_Y(y) &= \frac{dG_Y(y)}{dY} = \frac{dG_Y(y)}{dX}\frac{dX}{dY} = \frac{d(F(\frac{Y}{2\sqrt{2b+1}} + \frac{1}{2}))}{dX}\frac{dX}{dY} = \\ &= f_X\left(\frac{Y}{2\sqrt{2b+1}} + \frac{1}{2}\right)\frac{d(\frac{Y}{2\sqrt{2b+1}} + \frac{1}{2})}{dY} = \\ &= \frac{1}{2\sqrt{2b+1}} f_X\left(\frac{Y}{2\sqrt{2b+1}} + \frac{1}{2}\right) \Rightarrow \end{aligned}$$

$$\begin{aligned} g_Y(y) = & \frac{1}{2\sqrt{2b+1}}\frac{\Gamma(2b)}{\Gamma^2(b)}\left(\frac{1}{4} - \frac{y^2}{4(2b+1)}\right)^{b-1}, \\ & -\sqrt{2b+1} < y < \sqrt{2b+1}. \end{aligned} \tag{A.1}$$

Stirling's approximation gives an approximate value for the gamma function $\Gamma(n)$ for $n \to \infty$:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

Therefore:

$$\Gamma(b) = \frac{b!}{b} \approx \frac{\sqrt{2\pi b}}{b}\left(\frac{b}{e}\right)^b = \sqrt{\frac{2\pi}{b}}\left(\frac{b}{e}\right)^b,$$

and:

$$\Gamma^2(b) \approx \frac{2\pi}{b}\left(\frac{b}{e}\right)^2 b. \tag{A.2}$$

Moreover:

$$\Gamma(2b) = \frac{2b!}{2b} \approx \frac{\sqrt{2\pi 2b}}{2b}\left(\frac{2b}{e}\right)^{2b} = 2^{2b}\sqrt{\frac{\pi}{b}}\left(\frac{b}{e}\right)^{2b}. \tag{A.3}$$

Substituting Eqs. (A.2) and (A.3) in Eq. (A.1), we obtain:

$$\begin{aligned} g_Y(y) &= \frac{1}{2\sqrt{2b+1}}\frac{2^{2b}\sqrt{\frac{\pi}{b}}\left(\frac{b}{e}\right)^{2b}}{\frac{2\pi}{b}\left(\frac{b}{e}\right)^{2b}}\left(\frac{1}{4} - \frac{y^2}{4(2b+1)}\right)^{b-1} = \\ &= \frac{2^{2b-2}}{\sqrt{\frac{2b+1}{b}}\sqrt{\pi}}\frac{1}{2^{2b-2}}\left(1 - \frac{y^2}{2b+1}\right)^{b-1} = \frac{\left(1 - \frac{y^2}{2b+1}\right)^{b-1}}{\sqrt{\frac{2b+1}{b}}\sqrt{\pi}}. \end{aligned}$$

$$\begin{aligned} \lim_{b\to\infty} g_Y(y) &= \lim_{b\to\infty}\frac{\left(1 - \frac{y^2}{2b+1}\right)^{b-1}}{\sqrt{\frac{2b+1}{b}}\sqrt{\pi}} \\ &= \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}, \quad -\sqrt{2b+1} < y < \sqrt{2b+1}. \end{aligned}$$

Thus, this limit converges pointwise to the probability density function of a standard normal random variable when $b \to \infty$, $g_Y(y)$. So, by Scheff's theorem [42], the distribution of Y converges to the standard normal distribution.

References

[1] P.A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, C. Hervas-Martinez, Ordinal regression methods: survey and experimental study, IEEE Trans. Knowl. Data Eng. 28 (1) (2016) 127–146, doi:10.1109/TKDE.2015.2457911.

[2] Z. Ma, S. Chen, A convex formulation for multiple ordinal output classification, Pattern Recognit. 86 (2019) 73–84, doi:10.1016/j.patcog.2018.09.005.

[3] H. Zhao, Z. Wang, P. Liu, The ordinal relation preserving binary codes, Pattern Recognit. 48 (10) (2015) 3169–3179, doi:10.1016/j.patcog.2015.02.011.

[4] M. Pérez-Ortiz, P. Gutiérrez, M. Carbonero-Ruz, C. Hervás-Martnez, Semi-supervised learning for ordinal kernel discriminant analysis, Neural Netw. 84 (2016) 57–66, doi:10.1016/j.neunet.2016.08.004.

[5] V. Gulshan, L. Peng, M. Coram, M.C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al., Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs, JAMA 316 (22) (2016) 2402–2410, doi:10.1001/jama.2016.17216.

[6] Q. Tian, M. Cao, H. Sun, L. Qi, J. Mao, Y. Cao, J. Tang, Facial age estimation with bilateral relationships exploitation, Neurocomputing 444 (2021) 158–169, doi:10.1016/j.neucom.2020.07.149.

[7] J. Barbero-Gómez, P.-A. Gutiérrez, V.-M. Vargas, J.-A. Vallejo-Casas, C. Hervás-Martínez, An ordinal CNN approach for the assessment of neurological damage in Parkinsons disease patients, Expert Syst. Appl. (2021) 115271, doi:10.1016/j.eswa.2021.115271.

[8] D. Guijo-Rubio, C. Casanova-Mateo, J. Sanz-Justo, P. Gutierrez, S. Cornejo-Bueno, C. Hervás, S. Salcedo-Sanz, Ordinal regression algorithms for the analysis of convective situations over Madrid-Barajas airport, Atmos. Res. 236 (2020) 104798, doi:10.1016/j.atmosres.2019.104798.

[9] B. Abraham, M.S. Nair, Automated grading of prostate cancer using convolutional neural network and ordinal class classifier, Inf. Med. Unlocked 17 (2019) 100256, doi:10.1016/j.imu.2019.100256.

[10] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, doi:10.1038/nature14539.

[11] L. Fang, H. Zhang, J. Zhou, X. Wang, Image classification with an RGB-channel nonsubsampled contourlet transform and a convolutional neural network, Neurocomputing (2019) 1–12, doi:10.1016/j.neucom.2018.10.094.
[12] G. Song, Z. Wang, F. Han, S. Ding, M.A. Iqbal, Music auto-tagging using deep recurrent neural networks, Neurocomputing 292 (2018) 104–110, doi:10.1016/j.neucom.2018.02.076.
[13] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533, doi:10.1038/nature14236.
[14] W. Ma, Y. Wu, F. Cen, G. Wang, MDFN: multi-scale deep feature learning network for object detection, Pattern Recognit. 100 (2020) 107149, doi:10.1016/j.patcog.2019.107149.
[15] X. Yuan, P. He, Q. Zhu, X. Li, Adversarial examples: attacks and defenses for deep learning, IEEE Trans. Neural Netw. Learn. Syst. 30 (9) (2019) 2805–2824, doi:10.1109/TNNLS.2018.2886017.
[16] M. Madadi, H. Bertiche, S. Escalera, SMPLR: deep learning based SMPL reverse for 3D human pose and shape recovery, Pattern Recognit. 106 (2020) 107472, doi:10.1016/j.patcog.2020.107472.
[17] L. Li, X. Zhao, W. Lu, S. Tan, Deep learning for variational multimodality tumor segmentation in PET/CT, Neurocomputing 392 (2019) 1–19, doi:10.1016/j.neucom.2018.10.099.
[18] L. Wang, X. Qian, Y. Zhang, J. Shen, X. Cao, Enhancing sketch-based image retrieval by CNN semantic re-ranking, IEEE Trans. Cybern. 50 (7) (2019) 3330–3342, doi:10.1109/TCYB.2019.2894498.
[19] S.R. Dubey, A decade survey of content based image retrieval using deep learning, IEEE Trans. Circuits Syst. Video Technol. (2021), doi:10.1109/TCSVT.2021.3080920.
[20] T. Zhao, B. Zhang, M. He, W. Zhang, N. Zhou, J. Yu, J. Fan, Embedding visual hierarchy with deep networks for large-scale visual recognition, IEEE Trans. Image Process. 27 (10) (2018) 4740–4755, doi:10.1109/TIP.2018.2845118.
[21] X. Geng, Z.-H. Zhou, K. Smith-Miles, Automatic age estimation based on facial aging patterns, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2234–2240, doi:10.1109/TPAMI.2007.70733.
[22] J. Xing, K. Li, W. Hu, C. Yuan, H. Ling, Diagnosing deep learning models for high accuracy age estimation from a single image, Pattern Recognit. 66 (2017) 106–116, doi:10.1016/j.patcog.2017.01.005.
[23] G. Antipov, M. Baccouche, S.-A. Berrani, J.-L. Dugelay, Effective training of convolutional neural networks for face-based gender and age prediction, Pattern Recognit. 72 (2017) 15–26, doi:10.1016/j.patcog.2017.06.031.
[24] Y. Fu, T.S. Huang, Human age estimation with regression on discriminative aging manifold, IEEE Trans. Multimedia. 10 (4) (2008) 578–584, doi:10.1109/TMM.2008.921847.
[25] C. Beckham, C. Pal, Unimodal probability distributions for deep ordinal classification, in: Proceedings of the 34th International Conference on Machine Learning, vol. 70, 2017, pp. 411–419.
[26] J.F.P. da Costa, H. Alonso, J.S. Cardoso, The unimodal model for the classification of ordinal data, Neural Netw. 21 (1) (2008) 78–91, doi:10.1016/j.neunet.2007.10.003.
[27] X. Liu, F. Fan, L. Kong, Z. Diao, W. Xie, J. Lu, J. You, Unimodal regularized neuron stick-breaking for ordinal classification, Neurocomputing 388 (7) (2020) 34–44, doi:10.1016/j.neucom.2020.01.025.
[28] A. Agresti, An Introduction to Categorical Data Analysis, John Wiley & Sons, 2018.
[29] P. Wan Kai, Continuation-ratio model for categorical data: a Gibbs sampling approach, in: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1, 2008, pp. 1–6.
[30] M. Khan, S. Mohamed, B. Marlin, K. Murphy, A stick-breaking likelihood for categorical data analysis with latent gaussian models, in: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, 2012, pp. 610–618.
[31] C. Beckham, C. Pal, A simple squared-error reformulation for ordinal classification, arXiv preprint (2016) 1–8 1612.00775.
[32] L. Wang, J. Gu, Y. Chen, Y. Liang, W. Zhang, J. Pu, H. Chen, Automated segmentation of the optic disc from fundus images using an asymmetric deep learning network, Pattern Recognit. 112 (2021) 107810, doi:10.1016/j.patcog.2020.107810.
[33] J. de la Torre, D. Puig, A. Valls, Weighted kappa loss function for multi-class classification of ordinal data in deep learning, Pattern Recognit. Lett. 105 (2018) 144–154, doi:10.1016/j.patrec.2017.05.018.
[34] P. Li, Y. Hu, X. Wu, R. He, Z. Sun, Deep label refinement for age estimation, Pattern Recognit. 100 (2020) 107178, doi:10.1016/j.patcog.2019.107178.
[35] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[36] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, 2015, pp. 1–15. 11245/1.505367
[37] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48, doi:10.1186/s40537-019-0197-0.
[38] G. King, L. Zeng, Logistic regression in rare events data, Polit. Aanal. 9 (2) (2001) 137–163, doi:10.1093/oxfordjournals.pan.a004868.
[39] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, P.A. Gutiérrez, Metrics to guide a multi-objective evolutionary algorithm for ordinal classification, Neurocomputing 135 (2014) 21–31, doi:10.1016/j.neucom.2013.05.058.
[40] R.G. Miller Jr, Beyond ANOVA: Basics of Applied Statistics, Chapman and Hall/CRC, 1997.
[41] J.W. Tukey, Comparing individual means in the analysis of variance, Biometrics 5 (2) (1949) 99–114, doi:10.2307/3001913.
[42] H. Scheffé, A useful convergence theorem for probability distributions, Ann. Math. Stat. 18 (3) (1947) 434–438, doi:10.1214/aoms/1177730390.

Víctor Manuel Vargas was born in Córdoba, Spain. He received the degree of Computer Engineering from the University of Córdoba, Spain, in 2018. Then, he received his master's degree in artificial intelligence research in 2019 at the International University of Menéndez Pelayo, Spain. He is currently doing his Ph.D. thesis in advanced computing, energy and plasma. He is a member of the AYRNA group of the Department of Computer Science and Numerical Analysis of the University of Córdoba. His current research interests include deep neural networks and their applications to image classification on nominal and ordinal problems.

Pedro Antonio Gutiérrez received the B.S. degree in computer science from the University of Sevilla (Spain) in 2006, and the Ph.D. degree in computer science and artificial intelligence from the University of Granada (Spain) in 2009. He is currently an Assistant Professor with the Department of Computer Science and Numerical Analysis, University of Córdoba (Spain). His research interests are in the areas of supervised learning, evolutionary artificial neural networks, ordinal classification and the application of these techniques to different real world problems, including precision agriculture, renewable energy, climatology and biomedicine, among others. He serves on the Editorial board for the journals IEEE Transaction on Neural Networks and Learning Systems and Progress in Artificial Intelligence and on the organization/program committees of several computational intelligence conferences.

César Hervás-Martínez was born in Cuenca, Spain. He received the B.S. degree in statistics and operations research from the Universidad Complutense de Madrid, Madrid, Spain, in 1978, and the Ph.D. degree in mathematics from the University of Seville, Seville, Spain, in 1986. He is currently a Professor of Computer Science and Artificial Intelligence with the Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain, and an Associate Professor with the Department of Mathematics and Engineering, Loyola University Andalucía, Spain. His current research interests include neural networks, evolutionary computation, and the modelling of natural systems.