Probabilistic Deep Ordinal Regression Based on Gaussian Processes

Yanzhu Liu, Fan Wang, Adams Wai Kin Kong
Nanyang Technological University
50 Nanyang Avenue, Singapore, 639798
yzliu@ntu.edu.sg, fan005@e.ntu.edu.sg, adamskong@ntu.edu.sg

Abstract

With excellent representation power for complex data, deep neural networks (DNNs) based approaches are state-of-the-art for ordinal regression problem which aims to classify instances into ordinal categories. However, DNNs are not able to capture uncertainties and produce probabilistic interpretations. As a probabilistic model, Gaussian Processes (GPs) on the other hand offers uncertainty information, which is nonetheless lack of scalability for large datasets. This paper adapts traditional GPs for ordinal regression problem by using both conjugate and non-conjugate ordinal likelihood. Based on that, it proposes a deep neural network with a GPs layer on the top, which is trained end-to-end by the stochastic gradient descent method for both neural network parameters and GPs parameters. The parameters in the ordinal likelihood function are learned as neural network parameters so that the proposed framework is able to produce fitted likelihood functions for training sets and make probabilistic predictions for test points. Experimental results on three real-world benchmarks – image aesthetics rating, historical image grading and age group estimation – demonstrate that in terms of mean absolute error, the proposed approach outperforms state-of-the-art ordinal regression approaches and provides the confidence for predictions.

1. Introduction

Ordinal regression is a supervised learning problem aiming to predict discrete labels with a natural order. An example is apparent age group estimation, which grades face images based on an ordinal scale such as “Infants”, “Children”, “Teenagers”, “Youth”, “Young adults”, “Adults”, “Middle aged ” and “Aged”. Ordinal regression can be viewed as a special case of metric regression, where the regression targets are discrete and finite, and the differences between adjacent labels are not necessary to be equal. If the ordinal relationship of labels is ignored, the problem becomes to multi-class classification.

Deep neural networks (DNNs) have attracted great attention in these several years and performed well on many classification problems. There are a few works [16] [12] [1] [13] employing DNNs to ordinal regression problems. All of them transformed ordinal regression problems to certain classification problems by taking ordinal relationship between categories into consideration. If ordinal regression is viewed as a bridge between multi-class classification and metric regression, existing DNNs based ordinal regression approaches look it from the classification side. Therefore, they focused more on whether an instance belongs to a category or not rather than how close to its ground truth category. Taking advantages of representation power and scalability of deep learning, DNNs based ordinal regression approaches are state-of-the-art. However, they also inherit standard DNNs’ limitations: being not able to tell whether a model is certain about its output like probabilistic models.

Gaussian Process, as a probabilistic model, learns the distributions over functions and is able to offer confidence bounds for prediction. To benefit from both representative power and calibrated probabilistic modelling, efforts have been invested to combine DNNs and GPs recently ([5][6][3]). However, most of existing attempts worked as a separated fashion: DNNs are used to extract features and then traditional GPs are trained on the deep features [24]. An end-to-end model was proposed by Hinton & Salakhutdinov[8], but they used a large set of unlabelled data to pretrain deep belief networks (DBN) unsupervisedly, then fine-tuned the DBN + GP model with the limited labelled data. The main barrier blocking the combined model to be trained end-to-end for large dataset is that the performance of GPs cannot be guaranteed if it is optimized in stochastic mini-batch manner, especially for non-Gaussian likelihood. Hensman *et al*. [7] proposed a variational approach to allow stochastic optimization to GPs classification, [23] and [2] integrated DNNs to GPs for multi-class classification. To the best of our knowledge, there is no existing work to model ordinal data by DNNs and GPs hybrid networks.

This paper adapts GPs regression to ordinal regression

by involving a double sigmoid likelihood function, which is a non-conjugate likelihood. Chu *et al*. [4] used a Gaussian function as the ordinal likelihood mainly because it is convenient to calculate MAP or Expectation Propagation optimization, but losing the precise estimation to ideal ordinal likelihood. Following the variational approach in [7], the proposed GPs for ordinal regression, as a network layer embedded in a Convolutional neural network, is trained in stochastic mini-batch manner. The parameters in the likelihood function are also trained as network parameters. Therefore, the proposed approach produces fitted likelihood and uncertainty for predictions. The contributions of this paper are highlighted as follows:

1. To the best of our knowledge, the proposed approach is the first attempt of deep probabilistic model for ordinal regression applicable to large datasets.

2. It extends DNN and GPs hybrid network with stochastic optimization to both conjugate and non-conjugate ordinal likelihood.

3. The parameters in the ordinal likelihood function are learned as neural network parameters so that the proposed framework is able to produce fitted likelihood functions for training sets.

The rest of this paper is organized as follows. Section 2 reviews the literature of ordinal regression. Section 3 and 4 describe the proposed likelihood function and the network architecture. Section 5 reports the experimental results, and section 6 gives conclusive remarks.

2. Related Work

Niu *et al*. claimed that the adapted DNN in [16] is the first deep learning model for ordinal regression. For a m-rank ordinal regression problem, they constructed $m-1$ binary classifiers with the k-th one answering the question "Is the rank of an instance greater than k"? And a single CNN is used to combine all classifiers and output the $k-1$ predictions for an instance. The final prediction is decoded from these $k-1$ outputs. Liu *et al*. [12] focused more on small dataset ordinal regression, and they proposed to explore ordinal data relationship from triplets of instances through DNNs. An m-rank ordinal regression problem was transformed to m binary classification problems with triplets whose elements are from different ranks as inputs. The k-th classifier answered the question "Is the rank of an instance greater than $k-1$ and smaller than $k+1$?" m separate CNNs were used and the prediction for an instance was made by majority voting. In both approaches, a decoder was needed to recover the rank prediction from the outputs of ordinal classifiers. [13] proposed a constrained DNN for the ordinal regression (CNNPOR) which minimized multi-class classification loss with regression constraints keeping instances from different ranks in order. CNNPOR obtained predictions for instances without decoding and achieved state-of-the-art performance for ordinal regression in terms of zero-one error. All the above approaches tackle ordinal regression problem from classification perspective. They do not capture that how close the rank of an instance is to the ground truth. In other words, if an instance belongs to rank k, the probability that it is predicted as $k-1$ is not necessarily higher than that as $k-2$.

In the literature, there are handcrafted feature-based approaches solving ordinal regression problems from regression perspective. For an m-rank ordinal regression problem, it is assumed that there is a latent function mapping the instances to a real line. And there exists $m-1$ boundaries dividing the real line into m continuous intervals corresponding to the m categories of the problem. The targets of this type of ordinal regression approaches are to learn the mapping function and the boundaries. [4] (GPOR) proposed to extend GPs regression for ordinal regression. They used a Gaussian likelihood as an approximation to the ideal ordinal likelihood because the inference can be performed in closed form. Because of the expensive computation of solving GPs, GPOR was performed on small handcrafted feature datasets only.

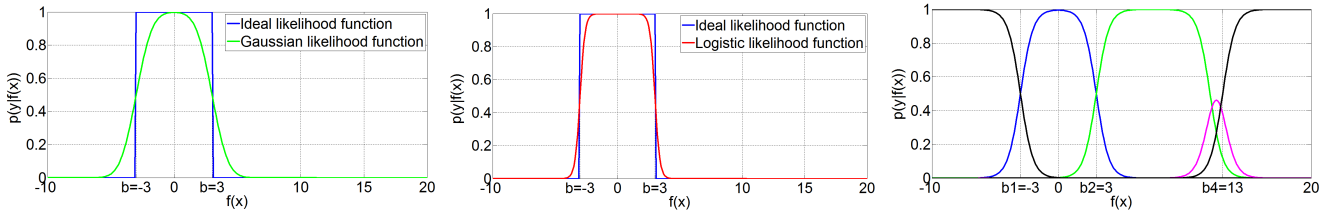
To reduce the computational cost of inference in GPs, numerous variational approaches have been proposed for GPs regression. Hensman *et al*. [7] showed a variational approach enabling GPs to be optimized in stochastic mini-batch manner if the likelihood is Gaussian. In [7], Hensman *et al*. extended the approach for multi-class classification, which has a non-conjugate likelihood (see section 3). This paper extends it further for ordinal regression with a non-conjugate likelihood.

3. Scalable Variational Gaussian Processes for Ordinal Regression

An ordinal regression problem with m ordinal ranks is considered. A training set with labeled instances $\mathcal{D} = \{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ is given, where $\mathcal{X}$ is the input space, and $\mathcal{Y} = \{1, 2, \cdots, m\}$ is the label space and the natural order of the numbers in $\mathcal{Y}$ indicates the order of the ranks. The target is to predict the rank label $y_t \in \mathcal{Y}$ for any new input $\mathbf{x}_t \in \mathcal{X}$.

3.1. Gaussian Processes Regression

Gaussian Process is a stochastic process that any finite sub-collection of random variables has a multivariate Gaussian distribution. It defines a distribution $p(\mathbf{f})$ over latent function $\mathbf{f}$. A zero-mean multivariate Gaussian distribution is assumed for the prior distribution of $\mathbf{f}$, $p(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{nn})$, where $\mathbf{K}_{nn}$ is the covariance matrix



(a) Gaussian likelihood function for one rank (b) Logistic likelihood function for one rank (c) Logistic likelihood functions learned for a 5-rank ordinal regression problem

Figure 1: Likelihood function examples

that $\mathbf{K}_{nn} = K(\mathbf{x}, \mathbf{x})$ and $K(\cdot, \cdot)$ is a kernel function. The likelihood function for Gaussian Process Regression is defined by $p(\mathbf{y}|\mathbf{f}) = \mathbf{f} + \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. With the Gaussian likelihood function, the marginal likelihood and predictive distribution given a new input vector $\mathbf{x}_*$ can be analytically derived as:

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{nn} + \sigma^2\mathbf{I}) \tag{1}$$

and

$$\begin{aligned} p(y_*|\mathbf{x}_*, \mathcal{D}) =& \mathcal{N}(y_*|\mathbf{K}_{n*}^\top(\mathbf{K}_{nn} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \\ & \mathbf{K}_{**} - \mathbf{K}_{n*}^\top(\mathbf{K}_{nn} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{n*}) \end{aligned} \tag{2}$$

respectively, where $\mathbf{K}_{n*}$ and $\mathbf{K}_{**}$ are covariance matrices $\mathbf{K}_{n*} = K(\mathbf{x}_*, \mathbf{x})$ and $\mathbf{K}_{**} = K(\mathbf{x}_*, \mathbf{x}_*)$.

3.2. Ordinal Likelihood

In ordinal regression, the noise-free likelihood is defined by:

$$p(y_i = k|f_i) = \begin{cases} 1, & \text{if } b_{k-1} < f_i \leq b_k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $-\infty = b_0 < b_1 < \cdots < b_{m-2} = \infty$. We use f_i to denote $f(\mathbf{x}_i)$. We then define $b_1 \in \mathbb{R}$, and for all $k \in \{2, \ldots, m-2\}$, $b_k = b_1 + \sum_{i=1}^{k-1} \Delta_i^2$ where $\Delta_i \neq 0$. Those $b_k's$ specify m intervals which map the real-valued $f(\mathbf{x}_i)$ into m categories in set $\mathcal{Y}$.

However, the noise-free likelihood is hardly used since it is not differentiable. With noise being considered, Chu and Ghahramani [4] proposed a likelihood function for ordinal regression by introducing a noise variable $\delta \sim \mathcal{N}(0, \sigma^2)$, which follows a Gaussian distribution with mean zero and variance σ^2. Hence the likelihood can be derived as:

$$\begin{aligned} p(y_i = k|f_i) &= \int p(y_i = k|f_i + \delta_i)\mathcal{N}(\delta_i; 0, \sigma^2)\, \mathrm{d}\delta_i \\ &= \Phi(\frac{b_k - f_i}{\sigma}) - \Phi(\frac{b_{k-1} - f_i}{\sigma}) \end{aligned} \tag{4}$$

where $\Phi(x) = \int_{-\infty}^{x} \mathcal{N}(\theta; 0, 1)\, d\theta$ is the cumulative density function of standard Gaussian distribution.

3.2.1 A Non-Conjugate Ordinal Likelihood

Besides Gaussian noise introduced above, we extend the likelihood for ordinal regression to the difference of any cumulative-density-like functions. In this paper, we only discuss logistic likelihood, which is closer to the noise-free likelihood, comparing with the Gaussian one (Figure 1). The method can be naturally extended to other likelihood functions. More specifically, the logistic likelihood can be written as:

$$p(y_i = k|f_i) = \mathrm{sig}(\frac{b_k - f_i}{\sigma}) - \mathrm{sig}(\frac{b_{k-1} - f_i}{\sigma}) \tag{5}$$

where $\mathrm{sig}(\cdot)$ is the sigmoid function, $\mathrm{sig}(x) = 1/(1 + e^{-x})$.

3.3. Scalable Variational Gaussian Process for Ordinal Regression

Gaussian Processes are inefficient for large datasets, suffering from its computational complexity of $\mathcal{O}(N^3)$ when training, which requires inverting the matrix $\mathbf{K}_{nn} + \sigma^2\mathbf{I}$, and $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$ for calculating mean and variance of predictive distribution based on the inversion. To solve this problem, Sparse Pseudo-input Gaussian Processes(SPGP) was first presented by Snelson and Ghahramani [21], and further modified as fully independent training conditional (FITC) approximation by Quiñonero-Candela and Rasmussen [18]. FITC introduces inducing points $\mathbf{Z}$ and $\mathbf{u}$, which is assumed that $\mathbf{u}$ and the latent function $\mathbf{f}$ have same form. Thus, the joint prior $p(\mathbf{f}, \mathbf{u})$ and marginal likelihood $p(\mathbf{y})$ are:

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \mathbf{K}_{nn}, \mathbf{K}_{nm} \\ \mathbf{K}_{nm}^\top, \mathbf{K}_{mm} \end{bmatrix} \right) \tag{6}$$

$$\begin{aligned} p(\mathbf{y}) &= \int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\, \mathrm{d}\mathbf{u} \\ &= \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{Q}_{nn} + \mathrm{diag}(\mathbf{K}_{nn} - \mathbf{Q}_{nn}) + \sigma^2\mathbf{I}) \end{aligned} \tag{7}$$

where matrices $\mathbf{K}_{nm} = K(\mathbf{x}, \mathbf{u})$, $\mathbf{K}_{mm} = K(\mathbf{u}, \mathbf{u})$ and $\mathbf{Q}_{nn} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{nm}^\top$

Finding inducing points and hyperparameters requires maximizing the marginal likelihood with respect to all the

parameters by gradient ascent [18]. In order to get a sparse approximation, Titsias *et al*. [22] proposed a variational approach which instead maximizes the lower bound of logarithm of the above marginal likelihood, with computation complexity only $\mathcal{O}(NM^2)$:

$$\log p(\mathbf{y}) \geq \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{Q}_{nn} + \sigma^2\mathbf{I}) - \frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{K}_{nn} - \mathbf{Q}_{nn}) \quad (8)$$

Note that the explicit lower bound for marginal likelihood is only tractable for conjugate case, e.g., Gaussian likelihood; for more general and non-conjugate cases, Hensman *et al*. [7] extended the above lower bound to:

$$\log p(\mathbf{y}) \geq -\mathcal{KL}[q(\mathbf{u})||p(\mathbf{u})] + \int q(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f})\, \mathrm{d}\mathbf{f} \quad (9)$$

where $q(\mathbf{f})$ is defined as $\int p(\mathbf{f}|\mathbf{u})q(\mathbf{u})\, \mathrm{d}\mathbf{u}$, which is a multivariate Gaussian distribution due to Gaussian assumption of $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{v}, \mathbf{C})$. Here the first term is simply a $\mathcal{KL}$ divergence term of two Gaussian distributions:

$$\mathcal{KL} = \frac{1}{2}\big[\log \frac{|\mathbf{K}_{mm}|}{|\mathbf{C}|} - m + \mathrm{tr}(\mathbf{K}_{mm}^{-1}\mathbf{C}) + \mathbf{v}^\top \mathbf{K}_{mm}^{-1}\mathbf{v}\big] \quad (10)$$

Since the likelihood in the integral $p(\mathbf{y}|\mathbf{f})$ can be factorized to the product of independent likelihood $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} p(y_i|f_i)$, the second term in Eq. 9 can be written as an expectation of log-likelihood with respect to a Gaussian, $\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] = \sum_{i=1}^{N} \mathbb{E}_{q(f_i)}[\log p(y_i|f_i)]$. For our case, the ordinal log-likelihood is:

$$\log p(y_i = k|f_i) = \frac{f_i}{\sigma} - \log(e^{-b_{k-1}/\sigma} - e^{-b_k/\sigma}) - \mathrm{LLP}(-\frac{b_k - f_i}{\sigma}) - \mathrm{LLP}(-\frac{b_{k-1} - f_i}{\sigma}) \quad (11)$$

where LLP refers to the logistic-log-partition function. The expectation of LLP is analytically intractable, but can be solved using the piecewise linear and quadratic bound of LLP [9][14], as well as the expectation of log-likelihood. Therefore, the lower bound in Eq. 9 is tractable and its gradient can be computed in analytical form.

The posterior distribution is derived based on the assumption that $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$. For any new testing vector $\mathbf{x}_*$, the posterior distribution of latent function $\mathbf{f}_*$ and $\mathbf{y}_*$is given by:

$$p(\mathbf{f}_*|\mathbf{x}_*, \mathcal{D}) = \int p(\mathbf{f}_*|\mathbf{u})q(\mathbf{u})\, \mathrm{d}\mathbf{u} \quad (12)$$

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathcal{D}) = \int p(\mathbf{f}_*|\mathbf{x}_*, \mathcal{D})p(\mathbf{y}_*|\mathbf{f}_*)\, \mathrm{d}\mathbf{f}_* \quad (13)$$

The integral in posterior distribution of $p(\mathbf{f}_*|\mathbf{x}_*, \mathcal{D})$ is Gaussian since both $p(\mathbf{f}_*|\mathbf{u})$ and $q(\mathbf{u})$ are Gaussian distributed. Its mean and variance can be computed in $\mathcal{O}(M^2)$, from which the posterior distribution of $p(\mathbf{y}_*|\mathbf{x}_*, \mathcal{D})$ is computed.

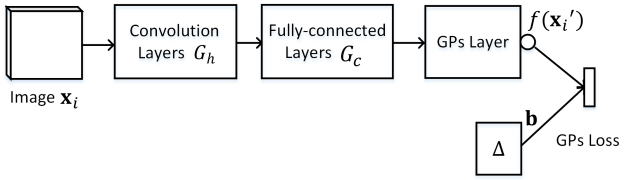


Figure 2: The architecture of GP-DNNOR.

4. A Probabilistic Deep Neural Network for Ordinal Regression (GP-DNNOR)

As GPs regression with ordinal likelihood can be optimized in stochastic mini-batch manner by using the method in section 3.3, a deep neural network for ordinal regression based on Gaussian Processes (GP-DNNOR) is proposed. Figure 2 shows the architecture of GP-DNNOR. Before the GPs layer, G_h and G_c in Figure 2 are standard convolution layers and fully connected layers in the classic CNNs, such as those in the VGG [20] network. G_h and G_c are learned to map an image $\mathbf{x}_i$ into high-level feature space, and the output vector is denoted as $\mathbf{x}_i'$. $\mathbf{x}_i'$ is the input of the GPs layer, *i.e*., the $\mathbf{x}$ in the covariance matrix $\mathbf{K}_{nn} = K(\mathbf{x}, \mathbf{x})$ of Eq. 1. In other words, during training of GP-DNNOR, both GPs parameters and GPs inputs are optimized simultaneously.

By using the variational approach in section 3.3, GPs layer provides predicted mean and variance $\mathbb{E}[f(\mathbf{x}_i')]$, $\mathbb{V}[f(\mathbf{x}_i')]$ for distribution of the latent function for $\mathbf{x}_i'$. We also make $b_k's$ trainable, by specifying a separate "Δ layer" with $m-2$ trainable nodes in Figure 2, which is designed to learn the boundaries $b = \{b_1, b_2, \cdots, b_{m-1}\}$. Being able to adjust the values of $b_k's$ along with the training process has been proven to be more efficient than pre-specified fixed $b_k's$ by experiments. To guarantee $b_k's$ satisfying the ordered constraints $b_{k-1} < b_k$, we define $b_k = b_1 + \sum_{i=1}^{k-1} \Delta_i^2$ as in Section 3.2 and represent positive $\Delta_i's$ as nodes in the Δ layer. To avoid shifting to $-\infty$, b_1 is fixed to -1 in the implementation. It should be pointed out that the value of b_1 does not affect the learning results, because as shown in Eq. 5, it is the value of $b_k - f_i$ that affects in the likelihood function, and f_i is automatically shifted according to b_1 during learning. One example of the benefits of trainable $b_k's$ can be observed from Figure 1c. Finally, a GPs loss layer which calculates the variational expectation loss in Eq. 9 is followed at the end.

For sake of clear presentation, the learning procedure of GP-DNNOR is summarized in Algorithm 1. The inputs are a training set with labeled instances and a set of testing images. The outputs include the predicted rank labels $\hat{y_j}$ and mean $\mathbb{E}[f(\mathbf{x}_j^*)]$ and variance $\mathbb{V}[f(\mathbf{x}_j^*)]$ of latent function values for the test images, and the boundaries b dividing the latent function values into ordinal intervals. The algo-

Algorithm 1 Pseudo code of learning in GP-DNNOR

Input: A training set $D = \{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i$ is an image and $y_i \in \{1, ..., m\}$ is its rank label. And a testing set $D_t = \{\mathbf{x}_j^*\}$.

Output: Predictions $Y_t = \{(\hat{y_j}, \mathbb{E}[f(\mathbf{x}_j^*)], \mathbb{V}[f(\mathbf{x}_j^*)])\}$ and boundary vector b where $b_1 = -1$, $b_{k+1} = b_k + \Delta_k^2$.

1: Initialize weights of G_h and G_c in Figure 2, and initialize $\Delta = \{\Delta_k = 1 | k = 1, \cdots, m-2\}$.
2: **for** $epoch = 1$ to MAX_{epoch} **do**
3: Shuffle D and divide it into mini-batches D_s.
4: **for each** D_s **do**
5: **procedure** FORWARD(D_s)
6: Forward propagate instances into G_h, G_c, and output vectors $\{\mathbf{x}_i'\}$ to the GPs layer.
7: Calculate $\{\mathbb{E}[f(\mathbf{x}_i')], \mathbb{V}[f(\mathbf{x}_i')]\}$ in the GPs layer and output to the loss layer.
8: Calculate the loss in Eq.9 using $\{\mathbb{E}[f(\mathbf{x}_i')]\}$ and b.
9: **end procedure**
10: Backward propagate through the GPs layer to G_c and G_h.
11: Update weights in G_c and G_h.
12: FORWARD(D_s)
13: Backward propagate to Δ.
14: Updates Δ.
15: **end for**
16: **end for**
17: Ouput b.
18: Forward propagate instances $\{\mathbf{x}_j^*\}$, and output $\{\mathbb{E}[f(\mathbf{x}_j^{*'})], \mathbb{V}[f(\mathbf{x}_j^{*'})])\}$ by the GPs layer.
19: Calculate $\{\hat{y_j}\}$ in Eq. 13 using $\{\mathbb{E}[f(\mathbf{x}_i^{*'})]\}$.

rithm starts with initialization; in experiments, the weights of G_h and G_c are initialized by VGG weights pretrained on ImageNet. At each training epoch, the training instances are randomly shuffled and redivided into mini-batches. For each mini-batch, two separate rounds of forward and backward propagations for f and b are performed as shown in lines $5-11$ and $12-14$. Instead of updating them simultaneously, we choose to update f and b, conditioning on fixed b and f, from the previous iteration, respectively. This is to guarantee that we optimize both conditional loss functions in the correct direction. In the forward procedure, GPs layer assigns the inducing points and calculate $\mathbb{E}[f(\mathbf{x}_j')]$ and $\mathbb{V}[f(\mathbf{x}_j')]$ (line 7). The number of inducing points in the experiments of this paper is set to the size of mini-batch. The loss to be minimized is the bound in Eq. 9 with the logistic ordinal likelihood with mean $\mathbb{E}[f(\mathbf{x}_j')]$ and b as inputs. The gradient of the loss with respect to $\mathbb{E}[f(\mathbf{x}_j')]$ is calculated and backpropagated through the GPs layer to G_c and G_h in line 10. All the weights in G_h and G_c are updated when b is kept fixed as the previous value; afterwards, the instances in the current mini-batch D_s are forward propagated again to update b. As shown in Algorithm 1, standard stochastic backprobagation is used. GP-DNNOR, therefore, is scalable for large scale datasets.

5. Evaluation

Different ordinal datasets contain varying degrees of ordinal information. Some of them are more close to classification while others are close to regression. Therefore, three datasets with very diverse images are employed in the experiments. Moreover, they are used to evaluate how GP-DNNOR performs on datasets with different data sizes, the number of ranks and class distributions.

- Image aesthetic benchmark [19] is to evaluate algorithms rating images to five aesthetic grades: *Unacceptable*, *Flawed*, *Ordinary*, *Professional* and *exceptional*. The "urban" category of the benchmark is used in the experiments, and Figure 3 shows an example image of each rank. The images are Flickr photos which are labeled to one of the five aesthetics levels. The total number of images is 3492 and the samples in different ranks are imbalanced. The second column of Table 1 lists the data size of each rank in this dataset. All approaches are tested on five random training/testing partitions following those in [13] for fair comparison.

- Historical image benchmark [17] stores color images photographed on five decades, *1930s* to *1970s*. Figure 4 shows some sample images. It is used to evaluate algorithms predicting the dates when images were photographed in terms of decade. The five decades correspond to five ordinal categories and each category has 265 images. This benchmark is a small scale ordinal dataset with balanced number of samples in each rank. The experiments are performed on 20 folds and the mean values of results are reported.

Table 1: Class distributions of the three datasets

#Images	Urban image aesthetics	Historical color images	Adience face
Rank 1	2	265	2293
Rank 2	135	265	1971
Rank 3	2203	265	1963
Rank 4	1003	265	1541
Rank 5	149	265	4530
Rank 6			2108
Rank 7			762
Rank 8			798
Total	3492	1325	15966

Figure 3: Urban image aesthetics dataset (*unacceptable* to *exceptional* from left to right)

Figure 4: Historical color image dataset (*1930s* to *1970s* from left to right)

Figure 5: Adience face dataset (*0-2* to *>60* from left to right)

- Adience face benchmark [10] has 15966 face images of eight age groups: *0-2*, *4-6*, *8-13*, *15-20*, *15-20*, *25-32*, *38-43* and *elder than 60 years old*. The images data in different age groups is imbalanced. Figure 5 shows example face images, and Table 1 lists the class distributions in the fourth column. This dataset is employed to evaluate the scalability of GP-DNNOR and the performance on relative large number of ranks with imbalanced training samples. In the experiments, the five-fold partition follows [10] for fair comparison.

GP-DNNOR is implemented in TensorFlow, and the GPs layer is implemented using GPflow [15]. The operations of GP layer to calculate from $\mathbf{x}'$ to $\mathbb{E}(\mathbf{x}')$ is a standard GP regression by introducing a latent variable $\mathbf{Z}$ as pseudo-inputs to support batch-based calculation. All deep methods employ VGG-16 as the basic architecture, and the convolutional layers and fully-connected layers are fine-tuned from the weights pretrained on ImageNet. In the training of GP-DNNOR for all the three datasets, the size of mini-batch is 150 and the learning rates for both f and b are 0.0001. The number of neurons in the last fully connected layer of GP-DNNOR is 100, and the number of inducing points is set to 150. In the experiments, the images are resized to 256×256 pixels and are randomly cropped to 224×224 pixels further during the learning.

In the rest of this section, we first compare GP-DNNOR with other probabilistic and non-probabilistic model, then derive the importance of uncertainty information provided by GP-DNNOR. We adapt Mean Absolute Errors (MAE) and accuracy as the comparing metrics. Note that accuracy is not quite adequate for evaluating ordinal regression, since it only counts the number of correctly classified samples and neglected ordinal errors. If two methods A and B assign an input sample to the third rank and the tenth rank but its ground truth label is the second rank, their errors in terms of accuracy are exactly the same. MAE is considered more suitable to evaluate ordinal regression, which puts more weights on measuring the distance between the true and the predicted rank.

5.1. Results Comparing with Probabilistic Models

To the best of our knowledge, GP-DNNOR is the first attempt to introduce probabilistic properties into the deep ordinal regression. We compare our method with CNNm-GP[2], which is an end-to-end model with GP classification layer on the top of a DNN. CNNm-GP focuses more on multi-class classification, whose top layer uses multi-class inverse-link likelihood function[7], while GP-DNNOR is designed for ordinal regression, with more complicated ordinal likelihood. Furthermore, as in Figure 2, GP-DNNOR includes a Δ layer to learn decision boundaries $b = \{b_1, b_2, \ldots, b_{m-1}\}$ as discussed in Section 4.

Experiments evaluating probabilistic methods were conducted on the above mentioned three datasets. Table 2 summaries the results in terms of MAE and accuracy.

Table 2: Results comparing with probabilistic models

	Urban image aesthetics		Historical Color Image		Adience Face	
	Accuracy (%)	MAE	Accuracy (%)	MAE	Accuracy (%)	MAE
CNNm-GP[2]	63.17	0.41	**49.98±2.90**	0.84±0.08	46.1±10.0	0.78±0.11
GP-DNNOR	**68.29**	**0.32**	46.60±2.98	**0.76±0.05**	**57.4±5.5**	**0.54±0.07**

For image aesthetic and Adience Face datasets, our proposed GP-DNNOR outperforms CNNm-GP in both accuracy and MAE, while for the historical color image dataset, it achieves 3.4% lower accuracy but better MAE. We note that historical color image dataset is extremely difficult as a classification problem that untrained human annotators only achieved 26% accuracy[17]. Due to its property of blurry boundaries among neighbouring ranks, GP-DNNOR, as an ordinal regression method which is trained to reduce the MAE, falls behind when classifying into the exact rank. It worth noting that samples misclassified by GP-DNNOR are likely to fall in the neighboring rank, while those misclassified by CNNm-GP are distributed more randomly.

It worth noting that there are some variations of CNNm-GP and GP-DNNOR, *e.g.*, the separately and jointly trained CNN and GP with different likelihood functions. Although those variations did not achieve the expected experimental results as GP-DNNOR, they provided significant perspective on the contributions of different components of the proposed method. When training CNN and GP separately with Gaussian and Logistic ordinal likelihood on the image aesthetic dataset, the accuracy is 67.14%, 67.37%, and MAE is 0.35, 0.34, respectively. Similarly, training with Gaussian ordinal likelihood, instead of the proposed Logistic likelihood, achieved 67.32% accuracy and 0.34 MAE.

5.2. Comparing with Non-probabilistic Models

To comprehensively evaluate the proposed approach, GP-DNNOR is also compared with state-of-the-art deep ordinal regression methods, which contain no information about uncertainties. RED-SVM[11], Niu *et al.*'s method[16] and CNNPOR[13] are baseline methods on general ordinal regression, and the results of a traditional non-probabilistic and non-ordinal multi-class DNN (denoted by CNNm) are cited from [13].

Table 3 presents the results of comparing GP-DNNOR with non-probabilistic models. It is observed that GP-DNNOR achieves the best MAE performance consistently for all the three datasets. As for accuracy, GP-DNNOR achieves 0.8% and 3.52% lower than the best benchmark given by CNNPOR for the urban image aesthetics and historical color datasets respectively, but the best performance for Adience face dataset. It should be emphasized that the methods performing better than GP-DNNOR in terms of accuracy are not probabilistic and are not able to provide uncertainty information.

As a reference, the comparison of non-probabilistic models shows that GP-DNNOR exploits the advantages of ordinal regression by reducing MAE, and presents comparable performance in accuracy. Moreover, GP-DNNOR is also capable of interpret uncertainty information of the prediction, which we will further discuss its value in next section.

5.3. Analysis of Uncertainty

Like other Gaussian Processes based models, GP-DNNOR method provides probabilistic perspective of ordinal regression. It not only assigns a label for a testing sample according to posterior mean, but also provides information of the uncertainties, i.e., how likely the model would assign a testing sample to a certain label. As shown in Figure 6, given a group of testing samples, GP-DNNOR classifies samples with $b_3 < \mathbb{E}[f(\mathbf{x}_i)] \leq b_4$ into class 5. Samples with $\mathbb{E}[f(\mathbf{x}_i)]$ falling outside the interval $(b_3, b_4]$ are not given the correct labels. We distinguish the correctly classified and misclassified samples by different colours. In spite of this, it gives the credible intervals, showing the probability of any given sample being classified to each class. Unlike softmax classification methods, which calculate the probabilities according to the means, $\mathbb{E}[f(\mathbf{x}_i)]$, GP-

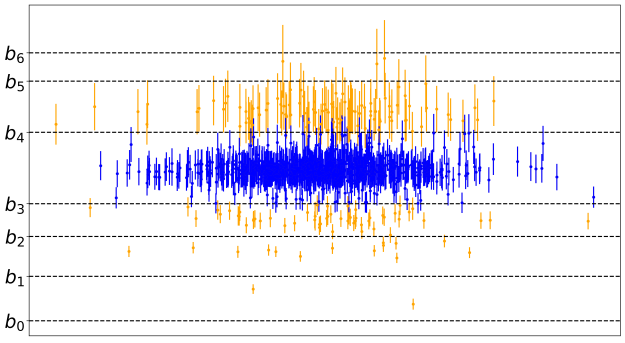


Figure 6: Predictions with uncertainties for Adience face dataset. Data samples being correctly classified into class 5 are highlighted in blue, while incorrect classifications to other classes are in orange.

Table 3: Results comparing with non-probabilistic models (SVM is not scalable to train on Adience Face dataset)

		Urban image aesthetics		Historical Color Image		Adience Face	
		Acc (%)	MAE	Acc (%)	MAE	Acc (%)	MAE
Classification	CNNm[13]	68.19	0.36	48.94±2.54	0.89±0.06	54.0±6.3	0.61±0.08
Ordinal regression	RED-SVM@8168[11]	63.88	0.39	35.92±4.69	0.96±0.06	-	-
	RED-SVM@deep[11]	65.44	0.37	25.38±2.34	1.08±0.05	-	-
	Niu *et al.*'s method[16]	66.49	0.35	44.67±4.24	0.81±0.06	44.7±4.2	0.81±0.06
	CNNPOR[13]	**69.09**	0.33	**50.12±2.65**	0.82±0.05	57.4±5.8	0.55±0.08
	GP-DNNOR	68.29	**0.32**	46.60±2.98	**0.76±0.05**	**57.4±5.5**	**0.54±0.07**

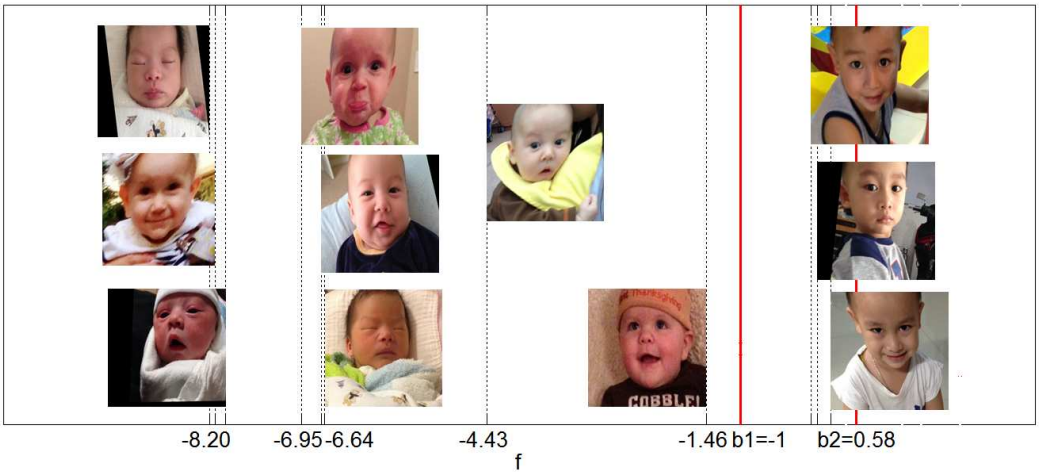


Figure 7: Test sample images of the first rank in Adience face dataset respects to the predicted f.

DNNOR instead gives a comprehensive overview through the posterior distribution. Based on the posterior distribution, we can easily derive the uncertainties of the results. Those error bars showing in Figure 6 are plotted using 95% credible intervals. Figure 6 shows an interesting property. For the correctly classified samples, the ratio of the error bar and the length of the interval is likely smaller than the ratio calculated from in correctly classified samples.

Besides, another advantage of GP-DNNOR over other deep learning ordinal methods, which are derived from the classification perspective, is that GP-DNNOR is able to interpret the continuous properties of datasets. Other deep learning ordinal methods assign sample to one of the labels, whereas GP-DNNOR captures the information in the middle of different classes. With the posterior distribution of the first rank (age group 0-2) in the Adience Face Dataset, GP-DNNOR also tells how likely the testing face is classified to 0-2 group, and neighbouring groups, along with age group assignment. In particular, within the group 0-2, GP-DNNOR is able to decide which subjects are close to age 0 and which are close to age 2, i.e., we can observe a linear relationship between the value of $\mathbb{E}[f(\mathbf{x}_i)]$ and exact ages of testing faces, which is not provided by the dataset. Figure 7 shows sample faces from 0-2 group. The red lines indicate the intervals and all the faces in Figure 7 are from group 0-2. Within the first interval, $(-\infty, b_1]$, it is observed that babies close to the left hand side are younger than those close to b_1. In the second interval $(b_1, b_2]$, the faces are also from 0-2 group, but they look much older than those in the first interval. Figure 7 shows that $\mathbb{E}[f(\mathbf{x}_i)]$ can capture information within and between the intervals.

6. Conclusions

This paper integrates a deep neural network and a GPs regression layer with non-conjugate likelihood for ordinal regression problems. The proposed network is trained end-to-end in the stochastic mini-batch manner, and the ordinal regression function and boundaries dividing it to ordinal intervals are learned simultaneously. The experimental results from the three datasets show that the proposed method achieves best MAE performances, and comparable accuracies to state-of-the-art methods. Moreover, it provides uncertainty information of predictions, which brings fresh insights into ordinal regression.

References

[1] Christopher Beckham and Christopher Pal. Unimodal probability distributions for deep ordinal classification. *arXiv preprint arXiv:1705.05278*, 2017.

[2] John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

[3] Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481, 2016.

[4] Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. In *Journal of Machine Learning Research*, pages 1019–1041, 2005.

[5] Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

[6] James Hensman and Neil D Lawrence. Nested variational compression in deep gaussian processes. *arXiv preprint arXiv:1412.1370*, 2014.

[7] James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable Variational Gaussian Process Classification. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 351–360, San Diego, California, USA, 09–12 May 2015. PMLR.

[8] Geoffrey E Hinton and Ruslan R Salakhutdinov. Using deep belief nets to learn covariance kernels for gaussian processes. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1249–1256. Curran Associates, Inc., 2008.

[9] Mohammad Khan. *Variational learning for latent Gaussian model of discrete data*. PhD thesis, University of British Columbia, 2012.

[10] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–42, 2015.

[11] Hsuan-Tien Lin and Ling Li. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367, 2012.

[12] Yanzhu Liu, Adams Wai Kin Kong, and Chi Keong Goh. Deep ordinal regression based on data relationship for small datasets. In *IJCAI*, pages 2372–2378, 2017.

[13] Yanzhu Liu, Adams Wai Kin Kong, and Chi Keong Goh. A constrained deep neural network for ordinal regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2018.

[14] Benjamin M. Marlin, Mohammad Emtiyaz Khan, and Kevin P. Murphy. Piecewise bounds for estimating bernoulli-logistic latent gaussian models. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 633–640, USA, 2011. Omnipress.

[15] De G Matthews, G Alexander, Mark Van Der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. Gpflow: A gaussian process library using tensorflow. *The Journal of Machine Learning Research*, 18(1):1299–1304, 2017.

[16] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4920–4928, 2016.

[17] Frank Palermo, James Hays, and Alexei A Efros. Dating historical color images. In *European Conference on Computer Vision*, pages 499–512. Springer, 2012.

[18] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[19] Rossano Schifanella, Miriam Redi, and Luca Maria Aiello. An image is worth more than a thousand favorites: Surfacing the hidden beauty of flickr pictures. In *ICWSM*, pages 397–406, 2015.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

[22] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

[23] Andrew G Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016.

[24] Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. Deep gaussian process for crop yield prediction based on remote sensing data. In *AAAI*, pages 4559–4566, 2017.