A Constrained Deep Neural Network for Ordinal Regression

Yanzhu Liu, Adams Wai Kin Kong
Nanyang Technological University
50 Nanyang Avenue, Singapore, 639798
liuy0109@e.ntu.edu.sg, adamskong@ntu.edu.sg

Chi Keong Goh
Rolls-Royce Advanced Technology Centre
6 Seletar Aerospace Rise, Singapore, 797575
ChiKeong.Goh@Rolls-Royce.com

Abstract

Ordinal regression is a supervised learning problem aiming to classify instances into ordinal categories. It is challenging to automatically extract high-level features for representing intraclass information and interclass ordinal relationship simultaneously. This paper proposes a constrained optimization formulation for the ordinal regression problem which minimizes the negative loglikelihood for multiple categories constrained by the order relationship between instances. Mathematically, it is equivalent to an unconstrained formulation with a pairwise regularizer. An implementation based on the CNN framework is proposed to solve the problem such that high-level features can be extracted automatically, and the optimal solution can be learned through the traditional back-propagation method. The proposed pairwise constraints make the algorithm work even on small datasets, and a proposed efficient implementation make it be scalable for large datasets. Experimental results on four real-world benchmarks demonstrate that the proposed algorithm outperforms the traditional deep learning approaches and other state-of-the-art approaches based on hand-crafted features.

1. Introduction

Ordinal regression, also named as ordinal classification, lies between multi-class classification and metric regression. Its problem setting is exactly same as that of multi-class classification, which is to predict the category label for an input instance. However, the categories in the ordinal regression problem have ordinal relationship. Ordinal regression also can be viewed as a discrete version of metric regression, but the difference is that the number of categories in ordinal regression is finite and the distances between categories are undefined. An example of ordinal regression is movie rating, which grades movies based on an ordinal scale such as 1 star to 5 stars, and a movie with 4 stars has a better rating than those with 3 stars.

Recently, a number of machine learning approaches have been proposed for ordinal regression. Most approaches resolve the ordinal regression problem either from regression prospective or from classification prospective. The approaches from regression prospective aim to learn a function mapping the instances to a real line and predict multiple boundaries to discretize the mapped value. For example, the max-margin based approaches [19][4] adapted the support vector regression to predict contiguous boundaries splitting the ordinal classes. The approaches from classification prospective embed ordinal information between class labels into the traditional classification methods. For example, neural network based approaches [8][2] use different coding schemes to encode the ordinal information of class labels into the output vectors of the networks. However, few of existing work combine classification and regression parts in the optimization objective explicitly.

In literature, most of existing ordinal regression approaches are based on handcrafted features, which are labor-intensive and highly rely on the prior knowledge. In these several years, deep neural networks (DNNs) have attracted great attention due to being able to automatically extract high-level features from raw data and performing very well on many classification tasks. However, very few works use DNNs for ordinal regression problem. Niu et al. (2016) [16] claimed that their method is the first work to adapt DNNs for ordinal regression. Generally speaking, a large training dataset is necessary to train a deep neural network, but many real-world ordinal regression problems are in fact small data problems. For example, for disease rating in medical images, in many cases large training image sets are not available because collecting such data is difficult, expensive and invasive. Learning deep neural networks on small datasets is challenging, and to design a method suitable for small datasets and also scalable for large datasets at the same time is another challenging task.

In the traditional deep learning approaches, the learning objectives are formulated as unconstrained optimization problems. Usually, the objective function is a loss designed for the task. This paper first formulates the ordinal regression problem as a constrained optimization problem and an

equivalent unconstrained formulation is derived. Based on this formulation, CNN can be adapted to solve it. A CNN can be viewed as a combination of multiple convolution layers which map instances to a high dimensional feature space and multiple fully-connected layers which perform as a classifier. The proposed method aims to learn the mapping function to maximize the probability of training instances belong to their category under constraints that in the high dimensional space, the instances from ordered ranks are mapped to a real line in order. The proposed constrained optimization problem can be converted to an unconstrained problem with two terms in the objective function: one is a logistic regression loss for classification and the other is a pairwise hinge loss for regression. Therefore, the proposed approach optimizes classification and regression objectives simultaneously, which targets to the problem setting of ordinal regression more directly.

The contributions of this paper are summarized as following: 1) The proposed approach adapts DNNs to solve a constrained optimization problem for ordinal regression. 2) The proposed approach is an end-to-end approach without any preprocessing such as feature extraction or any postprocessing such as decoding for predictions. 3) The proposed pairwise regularizer makes deep learning on small datasets possible. 4) The proposed approach is suitable for small datasets and scalable for large datasets.

The rest of this paper is organized as follows. Section 2 reviews the literature of ordinal regression. Section 3 describes the proposed objective function and the CNN architecture adapted for solving the optimization problem. Section 4 reports the experimental results. Section 5 gives conclusive remarks.

2. Related Work

As the problem setting of ordinal regression lies between multi-class classification and metric regression, the approaches in the literature can be divided into two categories: approaches from regression prospective and from classification prospective. In Gutierrez et al.'s survey [7], the first category of approaches is named as threshold approaches. These approaches assume that there is a latent function mapping the instances to a real line, and the ranks of instances are intervals on the line. The target of the threshold approaches is to learn the latent function and the boundaries of the intervals. For example, SVOR [4], a SVM based method, estimates the weight w for input vectors x and boundaries b, and the decision criteria is that the rank of x is k if and only if $w^T x \in [b_{k-1}, b_k]$, where b_k is the boundary separating rank k and rank $k+1$. Another method GPOR [3] assumes that the latent function $f(x)$ is a Gaussian process. A likelihood function $p(y|f(x))$ is proposed for ordinal regression and the hyperparameters including the boundaries b are estimated by MAP or EP algorithms. The threshold approaches are not able to predict the rank labels directly from the learned latent function without the boundaries b, but it is challenging to learn the boundaries especially when the number of ranks is large.

The second category of the approaches transform the ordinal regression problem to classification problems. Frank and Hall [6] propose to address the m-rank ordinal regression problem by using $m-1$ standard binary classifiers, and the k-th classifier is trained to predict the probability of the rank $y_t > k$ for an instance x_t. Then all the outputs from $m-1$ binary classifiers are combined to produce the decision of the rank of x_t. Cardoso and Pinto da Costa [1] and Li and Lin [12] propose two similar data replication methods independently to convert the ordinal regression problem to a binary classification problem. The RED-SVM approach in [12] extends a labeled instance (x, y) to $m-1$ binary instances for a m-rank ordinal regression problem by transformation: $x^k = (x; e^k) \in R^{d+m-1}$; $y^k = 1 - 2[y \leq k]$, where d is the dimension of x, $e^k \in R^{m-1}$ is an indicator vector for rank k, and $[\cdot]$ is the indication function. Based on the new dataset, a single binary classifier $f(x^k)$ is trained based on SVM to answer the question "Is the rank of an instance greater than k?". And the rank of x_t is predicted as $\sum_{k=1}^{m-1} [f(x_t^k) = 1] + 1$.

All the above methods rely on handcrafted features. In recent years, deep learning has achieved great success on classification problems, but there are very few works to apply DNNs on ordinal regression problems. Niu et al. [16] have recently adopted CNN for age estimation. They transform the m-rank ordinal regression problem to $m-1$ binary classifiers and the k-th classifier answers the question "Is the rank y_t of an instance greater than k"? The idea is very similar to RED-SVM, but they adapt a single CNN to combine all classifiers and output the $k-1$ predictions at the same time. However, a post-processing step is required to decode the final predicted rank for a testing instance x_t from possible contradictory outputs. For example, the outputs of the CNN predicts that y_t is greater than $k+1$ and smaller than $k-1$. In [16], Niu et al. follow the decoding strategy of RED-SVM to assign $y_t = \sum_{k=1}^{m-1} [f_k(x_t) = 1] + 1$, where $f_k(x_t)$ is the k-th output of the CNN for x_t.

In terms of ranking order, a related research topic is learning to rank for information retrieval. Its target is to learn the relevance between a document and a given query, and to predict the relative order of the documents based on the relevance. However, learning to rank is different from ordinal regression because it is not able to predict the exact ranks of documents. A comprehensive survey [13] summarizes the approaches of learning to rank as pointwise, pairwise and listwise approaches. RankingSVM [9], an pairwise approach, introduced the ranking constraints into SVM

as shown in Eq. 1:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2} \parallel w \parallel_2^2 + C \sum_{i,j} \xi_{i,j} \\ \text{s.t.} \quad & w \cdot \phi(q, d_i) \geq w \cdot \phi(q, d_j) + 1 - \xi_{i,j} \\ & \xi_{i,j} \geq 0 \end{aligned} \tag{1}$$

where q is a query, d_i and d_j are two documents, w is the weight vector and $\phi(q, d_i)$ is a mapping function. This paper adapts the pairwise constraints in Eq. 1 for ordinal regression and solves the proposed optimization problem under the deep learning framework.

3. The Proposed Algorithm

An ordinal regression problem with m ranks denoted by $Y = \{1, 2, \cdots, m\}$ is considered, where the natural order of the numbers in Y indicates the order of the ranks. A training set with labeled instances $T = \{(x_i, y_i) | x_i \in X, y_i \in Y\}$ is given, where X is the input space. The target is to predict the rank $y_t \in Y$ of an input $x_t \in X$. Let $X_k \subseteq X$ be the subset of training instances whose rank labels are k and $I_k = \{i | x_i \in X_k\}$ be the index set of X_k. Denote $x_i^k \in X_k$ as an input from rank k. In the rest of this section, the outline of the proposed approach will be provided first, and then the DNN architecture used to solve the optimization problem will be presented.

3.1. The Proposed Optimization Formulation

The intuition of the proposed approach is to learn a multi-class classifier by constraining the instances being mapped to a real line in order. Eq. 2 shows the optimization problem:

$$\begin{aligned} \min_{f,\phi,w,\xi} \quad & -\sum_{k=1}^{m} \sum_{i \in I_k} \log \frac{e^{f_k \circ \phi(x_i^k)}}{\sum_{r=1}^{m} e^{f_r \circ \phi(x_i^k)}} + C \sum_{k=1}^{m-1} \sum_{\substack{i \in I_k \\ j \in I_{k+1}}} \xi_{i,j}^k \\ \text{s.t.} \quad & w \cdot \phi(x_j^{k+1}) - w \cdot \phi(x_i^k) \geq 1 - \xi_{i,j}^k, \\ & \xi_{i,j}^k \geq 0, \quad k = 1...m-1, i \in I_k, j \in I_{k+1} \end{aligned} \tag{2}$$

where m is the number of ranks, and I_k is the index set of X_k. $f_k(\cdot)$ and $\phi(\cdot)$ are mapping functions, and $\circ$ is the function composition operator. w is the weight vector mapping $\phi(x)$ to a real line. $\phi(\cdot)$ can be considered as a feature extractor and $f_k(\cdot)$ is a classifier for label k. The first term of the objective function in Eq. 2 is the composition of softmax function and multinomial logistic regression loss, and the second term is the sum of slack variables $\xi_{i,j}^k$ where C is a hyperparameter. The constraints in Eq. 2 define the condition that the mapped values of instances from rank $k+1$ should be equal or larger than those of instances from rank k with an margin of 1 and tolerance $\xi_{i,j}^k$. Once the optimal solution of Eq. 2 is obtained, the rank label of a test instance x_t is predicted as the category with the maximum likelihood. More precisely, Eq. 3 is the decision function:

$$\hat{y_t} = \operatorname*{argmax}_k \frac{e^{f_k \circ \phi(x_t)}}{\sum_{j=1}^{m} e^{f_r \circ \phi(x_t)}} = \operatorname*{argmax}_k f_k \circ \phi(x_t) \tag{3}$$

The constraints in the proposed approach enforce that all pairs of instances from adjacent ranks are mapped in order with a tolerance, and they are similar to those in RankingSVM [9] as shown in Eq. 1. However, the proposed optimization problem is different from RankingSVM in the following four prospectives: 1) Given a query, the target of RankingSVM is to predict the order of test instances based on relevance. It is not able to predict the exact rank of a test instance. 2) The objective of RankingSVM is to minimize the margin (i.e, $\parallel w \parallel_2^2$) based on the large-margin theory in support vector regression. However, the objective of the proposed approach is to maximize the loglikelihood which is always used for classification problems. 3) The constraints of RankingSVM are applied to all possible pairs for a given query, but the proposed constraints applied to pairs of instances from adjacent ranks. 4) The mapping function $\phi(\cdot)$ in RankingSVM is predefined by a kernel function, but in the proposed approach $\phi(\cdot)$ is learned automatically by a deep neural network. In the proposed optimization problem, the constraints only count on pairs of instances from adjacent ranks, but other pairs of instances, such as instances from rank k and rank $k+2$, are not considered explicitly. The reason is that if both $(w \cdot \phi(x_1^k), w \cdot \phi(x_2^{k+1}))$ and $(w \cdot \phi(x_2^{k+1}), w \cdot \phi(x_3^{k+2}))$ are in order, it can be inferred that $(w \cdot \phi(x_1^k), w \cdot \phi(x_3^{k+2}))$ are also in order.

The slack variables in Eq. 2 and the slack variables in SVM have the same meaning. They both are used as tolerances for non-separable instances. In Eq. 2, if $w \cdot \phi(x_j^{k+1}) - w \cdot \phi(x_i^k) \geq 1$, the error $\xi_{i,j}^k$ should be 0. Otherwise, the error $\xi_{i,j}^k$ should be $1 - (w \cdot \phi(x_j^{k+1}) - w \cdot \phi(x_i^k))$. Therefore, the proposed optimization problem can be rewritten as an unconstrained optimization problem in Eq. 4.

$$\begin{aligned} \min_{f,\phi,w} \quad & -\sum_{k=1}^{m} \sum_{i \in I_k} \log \frac{e^{f_k \circ \phi(x_i^k)}}{\sum_{r=1}^{m} e^{f_r \circ \phi(x_i^k)}} + C \sum_{k=1}^{m-1} \sum_{\substack{i \in I_k \\ j \in I_{k+1}}} \max(0, \\ & 1 + w \cdot \phi(x_i^k) - w \cdot \phi(x_j^{k+1})) \end{aligned} \tag{4}$$

The first term in Eq. 4 is same as the the first term in Eq. 2 and the second term can be viewed as a pairwise hinge loss for regression. Therefore, the proposed approach optimizes the weighted combination of classification loss and regression loss explicitly, which directly represents the definition of ordinal regression problem.

3.2. The Proposed CNN based Optimization

Traditional feature based large-margin approaches often employ a function $\psi(x_i)$ mapping the input feature vector x_i to a high dimensional space. And a predefined kernel is used to represent the mapping function based on the kernel trick. The form of the kernel function and its hyperparameters affect the performance a lot. Deep neural networks are able to learn the high level features and weights of classifiers simultaneously. Therefore, a deep neural network is

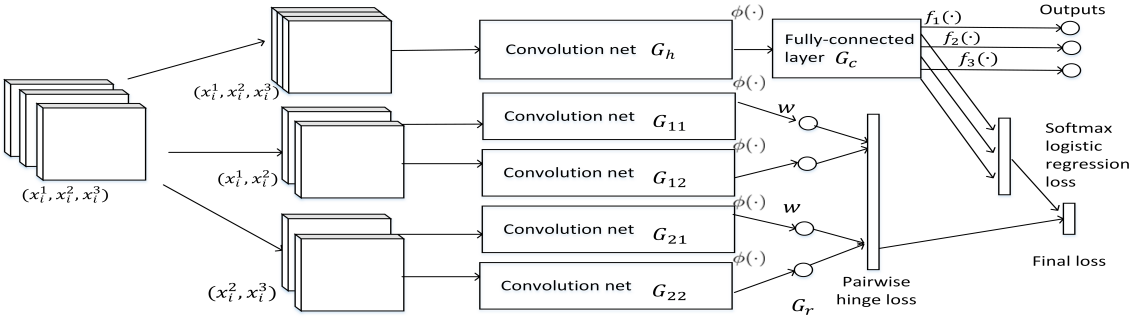

Figure 1: The architecture of CNNPOR for a 3-rank ordinal regression problem.

designed to learn the mapping function $\phi(\cdot)$, the weight w and $f_k(\cdot)$ in the proposed optimization problem in Eq. 4. Since convolutional neural networks are used in the current implementation, the proposed method is named convolutional neural network with pairwise regularization for ordinal regression (CNNPOR).

The new loss function defined in Eq. 4 is implemented in CNNPOR, which is a weighted combination of a softmax logistic regression loss and a pairwise hinge loss. It should be pointed out that the scales of the two losses are not same. Therefore, a new training set is constructed by pairing up the instances from adjacent ranks, i.e, $X' = \{(x_s^k, x_s^{k+1}) | x_s^k \in X_k, x_s^{k+1} \in X_{k+1}, k = 1, ..., m-1\}$. Define $P_k = \{(x_s^k, x_s^{k+1})\}$, and $I_k^p = \{s | (x_s^k, x_s^{k+1}) \in P_k\}$ as the index set of P_k. All the elements x_s^k and x_s^{k+1} in the pairs are used as input. Using this training set, the two losses are scaled automatically i.e. Eq. 5.

$$\min_{f,\phi,w} -\sum_{k=1}^{m-1}\sum_{s\in I_k^p} \log \frac{e^{f_k\circ\phi(x_s^k)}}{\sum_{r=1}^{m} e^{f_r\circ\phi(x_s^k)}} - \sum_{s\in I_{m-1}^p} \log \frac{e^{f_m\circ\phi(x_s^m)}}{\sum_{r=1}^{m} e^{f_r\circ\phi(x_s^m)}} + C\sum_{k=1}^{m-1}\sum_{s\in I_k^p} \max(0, 1 + w\cdot\phi(x_s^k) - w\cdot\phi(x_s^{k+1})) \quad (5)$$

Fig. 1 shows the architecture of CNNPOR for a 3-rank ordinal regression problem. The input instances are organized in a list, as (x_i^1, x_i^2, x_i^3) in the figure, where x_i^1, x_i^2, x_i^3 are from rank 1, 2 and 3, respectively. They are individually inputted to the convolution net G_h, which represents the mapping function $\phi(\cdot)$ in Eq. 5. The outputs of G_h as the high dimensional features are passed to the fully-connected layer G_c, which represents the mapping function $f_k(\cdot)$. There is a softmax logistic regression loss and the number of output neurons equals to the number the ranks. The combination of the convolution net G_h and the fully-connected layer G_c is a standard multi-class CNN. Then the instances from adjacent ranks (i.e, x_i^1 and x_i^2, x_i^2 and x_i^3) are paired up and inputted into the convolution nets G_{11} to G_{22}. The outputs of all G_{11} to G_{22} are mapped into one dimensional space by the mapping vector w, and then the pairwise hinge loss layer receives all the outputs to calculate the last term in Eq. 5. The final loss layer sums up the two losses at weights 1:C. All the convolution nets (G_h,G_{11}-G_{22}) have the same architecture which consists of layers before the last fully-connected layer in a standard CNN, and they share the same weights. In the training phase, the standard backprobagation technique is used and the loss is back propagated to all the convolution nets. In the testing phase, a testing point x_t is inputted into G_h and the output of the G_c is the prediction. Therefore, CNNPOR is different from other pairwise methods such as Niu et al.'s method [16], RED-SVM [12] and Liu et al.'s method [14], because it is an end-to-end approach for ordinal regression, which does not require any postprocess step to achieve the predictions.

3.3. Scalability of the Proposed Algorithm

The proposed pairwise constraints as a regularizer make learning CNNPOR on small datasets possible, while the proposed architecture is also computationally feasible for large datasets. It should be emphasized that, for a training set with n images, the number of input images of CNNPOR is n not n^2. As shown in Fig. 1, all the convolution layers G_h, G_{11}, G_{12}, G_{21} and G_{22} share weights, meaning that there is only one unique standard CNN to be trained. The pairwise constraints which require quadratic number of operations are applied on the features inputted to the pairwise loss layer, not on the raw input images.

Algorithm 1 describes the implementation of one training iteration in CNNPOR, which reorganizes the instances as each batch having d images from each rank (i.e., set D^r in Algorithm 1) and n images from all ranks randomly (i.e., set D^c). The training set is shuffled per epoch to make instances in mini-batches random. Assume that a standard CNN structure such as the VGG [20] or the LeNet [10] is used. All layers before the last fully-connected layer are named as G_h, which also represents for G_{11} to G_{22} in Fig. 1, and the last fully-connected layer is named as G_c. In CNNPOR, one more fully-connected layer G_r with one output node is connected to G_h, and its weights are the w in Fig. 1. As shown in line 1-2 of Algorithm 1, all instances of D are propagated to G_h. Then the instances of D^c are propagated to G_c to calculate the softmax loss l_1 and the instances of D^r are propagated to G_r to calculate the pairwise hinge

Algorithm 1 Pseudo code of one training iteration in CNNPOR

Input: Training set $D = D^c \cup D^r$ with n instances in D^c and $m \times d$ instances in D^r, where $D^r = D_1 \cup D_2 \cdots \cup D_m$, $D_k \subseteq X_k$ and the size of D_k is d.
Output: Update the network weights.

1: Initialize or update all weights in a CNN consisting of convolution net G_h and two fully-connected layers G_c and G_r both connected to G_h.
2: Forward propagate all instances of D into G_h.
3: Forward propagate instances of D^c into G_c.
4: Calculate the softmax loss l_1 of D^c.
5: Forward propagate instances of D^r into G_r.
6: **procedure** PAIRWISEHINGELOSS
7: Initialize pairwise hinge loss $l_2 \leftarrow 0$.
8: $O_k \leftarrow$ the outputs of G_r for D_k.
9: **for** $k = 1$ to $m - 1$ **do**
10: $l_2 = l_2 + SUM(MAX(0, 1 + O_k - O_{k+1}))$
11: **end for**
12: **end procedure**
13: Backward propagate of $l_1 + C \times l_2$.

loss l_2 in line 6-12. Finally, the weighted loss $l_1 + C \times l_2$ is back propagated to the whole network. O_k in line 8 is a vector where each element is the one-dimensional output of G_r for one instance of D_k, i.e., $w \cdot \phi(x_s^k)$ in Eq. 5. The operations '$-$', MAX and SUM in line 10 are element-wise substraction, maximum and summation. Therefore, comparing to a standard m-class CNN, for a mini-batch with $n + m \times d$ instances, CNNPOR does not calculate the softmax loss for $m \times d$ instances but calculates the hinge loss for them by using $m - 1$ element-wise vector substraction, maximum and summation operations instead. Thus, although CNNPOR introduces the pairwise regularizer, by employing the proposed architecture and implementation, it is scalable for large scale datasets.

4. Evaluation

The proposed CNNPOR approach is evaluated on four benchmarks - a historical color image dataset [17], an image retrieval dataset MSRA-MM1.0 [21], an image aesthetic dataset [18] and the Adience Face Dataset [11]. Accuracy and mean absolute error are used as performance indexes. Accuracy is defined by $\frac{1}{|T|} \sum_{x_t \in T} [\hat{y}_t = y_t]$, where T is a testing set and $|T|$ is its size, $[\cdot]$ is the indicator function, y_t is the ground truth of x_t, and $\hat{y}_t$ is its predicted label. Mean absolute error (MAE) is defined by $\frac{1}{|T|} \sum_{x_t \in T} |\hat{y}_t - y_t|$.

Three baseline methods are employed for comparison: the state-of-the-art handcarfted feature based ordinal regression method - RED-SVM [12], the traditional CNN method for multi-class classification - CNNm and the CNN based ordinal regression method - Niu et al.'s method [16].

4.1. Results on the Historical Color Images Dataset

The historical color image dataset [17] is a benchmark to evaluate algorithms predicting when a historical color image was photographed in the decade scale. The dataset stores images collected from five decades, 1930s to 1970s corresponding to five ordinal categories, and each category has 265 images. Fig. 2 shows samples in the dataset. The evaluation protocol reported in [17] is taken in this study for fair comparison. In each category, 215 images are employed for training and the rest 50 images are for testing.

Table 1 lists the experimental results on the historical color image dataset. Besides the results of the three baseline methods, i.e, RED-SVM, CNNm and Niu et al.'s method, the results from the previous methods on this dataset are also reported. Palermo et al.'s method [17] and Martin et al.'s method [15] are proposed for this particular task, and Frank and Hall's method [6] and Cardoso and Pinto da Costa's method [1] are for general ordinal regression problems. Palermo et al. [17] designed 8168 features for this task. In the experiments, all handcrafted feature based methods listed in Table 1 use the same features for fair comparison. RED-SVM [12] is a state-of-the-art handcrafted feature based method for general ordinal regression problems. To evaluate the performance of CNNPOR achieved by the deep features and by the algorithm, CNNPOR is compared with RED-SVM with the inputs of the 8168 handcrafted features (RED-SVM@8168 in Table 1) and the deep features extracted from the traditional CNN which are the 512 dimensional output values before the first fully-connected layer in the VGG architecture [20] (RED-SVM@deep in Table 1).

The deep multi-class classification method (CNNm in Table 1) and the deep ordinal regression method (Niu et al.'s method in Table 1) are implemented for comparison. For the historical image dataset, the VGG architecture [20] is employed for CNNm, Niu et al.'s method and CNNPOR. For CNNPOR, as shown in Fig. 1, G_h and G_c are linked together and implemented through the VGG architecture, i.e., G_h consists of the thirteen convolution layers and the ReLU and pooling layers in between, and G_c includes the three fully-connected layers and the layers in between. The implementation of $G_{11} - G_{22}$ is same as G_h. The images in the historical image dataset are resized to 256×256 pixels. For all the three deep learning methods, the image size of the input layer is set to 224×224 3-channel pixels, and the input images are cropped further at random positions during the training phases for data augmentation. For each training/testing image partition, the last 5 images in the training set are used as the validation images, i.e., 210, 5 and 50 images respectively for training, validation and testing in

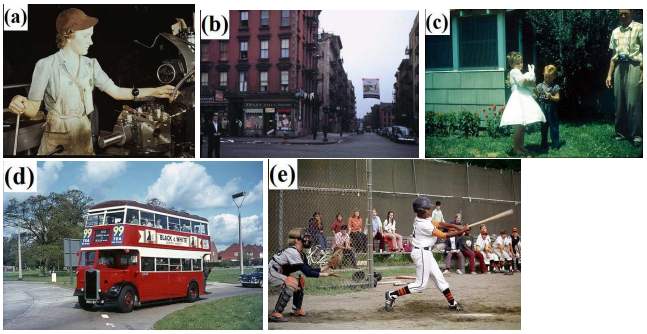

Figure 2: Historical color image dating dataset. (a)1930s, (b)1940s, (c)1950s, (d)1960s, (e)1970s.

Table 1: Results on the historical image benchmark.

Methods	Accuracy(%)	MAE
Palermo et al.'s method [17]	44.92±3.69	0.93±0.08
Martin et al.'s method [15]	42.76±1.33	0.87±0.05
Frank and Hall [6]	41.36±1.89	0.99±0.05
Cardoso and Pinto da Costa [1]	41.32±2.76	0.95±0.04
RED-SVM@8168 [12]	35.92±4.69	0.96±0.06
RED-SVM@deep [12]	25.38±2.34	1.08±0.05
CNNm	48.94±2.54	0.89±0.06
Niu et al.'s method [16]	44.67±4.24	**0.81±0.06**
CNNPOR	**50.12±2.65**	**0.82±0.05**

(a) Very relevant (b) Relevant (c) Irrelevant

Figure 3: MSRA-MM1.0 dataset: cat subset.

Table 2: Class distributions on MSRA-MM1.0 dataset.

	Rank 1	Rank 2	Rank 3	Total
Baby	379	295	277	951
Beach	336	398	213	947
Cat	243	344	378	965
Rose	222	418	329	969
Tiger	277	408	335	1020
Fish	130	669	165	964
Golf	777	97	79	953

each rank. In total, the sizes of training, validation and testing sets for CNNm and Niu et al.'s method are 1050, 25 and 250 images, respectively. For CNNPOR, all the possible permutations of the images in the five ranks produce $4 * 210^2$ training pairs (i.e., the pair (x_i^k, x_j^{k+1}) in Eq. 5) and $4 * 5^2$ validation pairs. All three deep methods are fine-tuned from the pretrained ImageNet model [20]. The C in Eq. 5 is set to 1 in the experiments. The learning rate of all layers, except for the last fully-connected layer, is set to 0.0001. Because the number of output nodes for the historical image dataset is different from that for the ImageNet, the learning rate of the last fully-connected layer is set as 10 times of the learning rate of other layers, i.e., 0.001.

Table 1 summarizes the results and the number after $\pm$ is the standard deviation values. CNNPOR outperforms RED-SVM on handcrafted features and deep features, CNNm, and Niu et al.'s method by 14.2%, 24.74%, 1.18%, and 5.45%, respectively in terms of accuracy. The mean MAE result of CNNPOR is 0.01 higher than that of Niu et al.'s method, which outperforms all other methods, but it is within two standard deviations of Niu et al.'s method. Overall, CNNPOR achieves the best results on the historical color image dataset. As shown in Table 1, CNNm performs much better than RED-SVM on deep features (RED-SVM@deep). The deep features for training RED-SVM are extracted from the well-trained CNNm. The results show RED-SVM cannot fully utilize the deep network, because during the training phase of RED-SVM, it cannot adjust the mapping from the raw images to deep features. As shown in Table 1, Niu et al.'s method achieves better performance for MAE than for accuracy. It is originally proposed for age estimation problem, which has larger number of ranks and is more similar to regression. Thus, it focuses on minimizing the absolute error, instead of zero one error.

4.2. Results on the Image Retrieval Dataset

Microsoft Research Asia Multimedia 1.0 (MSRA-MM 1.0) dataset [21] is a small scale benchmark which is constructed to evaluate multimedia information retrieval algorithms originally. MSRA-MM 1.0 has two parts, an image benchmark and a video benchmark. The image benchmark consists of 68 subsets. Each subset stores about 1000 images for one representative query from the image search engine of Microsoft Live Search. The images are thumbnails, i.e., the small images displayed on Microsoft Live Search. Fig. 3 shows a subset representing the query "cat". The relevance of the images to the corresponding query is classified into three levels: very relevant, relevant and irrelevant. Fig. 3 lists serval exemplar images labeled as "very relevant","relevant" and "irrelevant" to "cat". In the experiments, these three relevance levels are indicated by rank 1, 2 and 3. Given a testing image in a query set, we are targeting to predict which rank it belongs to. Seven subsets - "cat", "baby", "beach", "rose", "tiger", "fish" and "golf" in MSRA-MM 1.0 image benchmark are used to evaluate the performance CNNPOR. Table 2 summarizes the size of the

Table 3: Results on MSRA-MM1.0 dataset.

	Accuracy (%)					MAE				
	RED-SVM @8168	RED-SVM @deep	CNNm	Niu et al.	CNN-POR	RED-SVM @8168	RED-SVM @deep	CNNm	Niu et al.	CNN-POR
Baby	36.99	32.66	48.00	47.33	**50.00**	**0.630**	0.699	0.667	0.647	0.636
Beach	35.64	34.00	50.67	**51.11**	**51.11**	0.648	0.673	0.598	**0.576**	0.596
Cat	40.22	34.89	47.56	48.44	**52.89**	0.633	0.662	0.676	0.620	**0.598**
Rose	42.05	34.22	55.11	55.78	**56.67**	0.582	0.664	0.522	**0.500**	**0.500**
Tiger	35.57	33.56	**53.33**	51.78	52.89	0.644	0.673	0.571	**0.562**	0.578
Fish	68.66	**68.89**	63.95	66.16	66.33	0.313	**0.311**	0.378	0.357	0.355
Golf	80.45	80.17	83.08	83.93	**84.96**	0.283	0.289	0.229	0.219	**0.197**
Overall	48.51	45.48	57.39	57.79	**59.26**	0.533	0.567	0.520	0.497	**0.494**

(a) Unacceptable (b) Flawed (c) Ordinary (d) Professional (e) Exceptional

Figure 4: Image Aesthetics Dataset

seven subsets and the number of images in each rank. These datasets are small with less than 1100 images. To evaluate the algorithms on imbalanced datasets, "fish" and "golf" subsets are tested, respectively, 69.4% and 81.5% images in one rank. Besides images content and task differences, the images in MSRA-MM 1.0 are different from the historical images in three properties: different number of ranks, non-equal number of images in each rank or very imbalanced, and smaller image size.

Because the size of MSRA-MM 1.0 images is quite small, the LeNet architecture [10] is employed in all deep learning methods: CNNm, Niu et al.'s method and CNNPOR. The images are cropped to 60×60 pixels in the experiments. For each rank of the first five datasets in Table 2, the images are randomly split to 10 images for validation, 50 images for testing and the rest for training. For the two imbalanced datasets "fish" and "golf", 75%, 5% and 20% images in each rank are randomly selected for training, validation and testing, respectively. In each training set, 40960 pairs of instances from adjacent ranks are constructed as training instances for CNNPOR. Mini-batch size is set to 64 and the learning rate is set to 0.01. To evaluate RED-SVM method on handcrafted features, the same 8168 features as used for the historical image dataset are employed. RED-SVM is also tested on the features extracted before the first fully-connected layer of the LeNet architecture, which is 50 dimensional features. All methods are examined on three random training/testing partitions for all datasets and the mean results are summarized in Tables 3. CNNPOR performs better than all the baseline methods on five subsets in terms of accuracy, and on three subsets in terms of MAE. The results on MSRA-MM 1.0 dataset indicate that CNNPOR performs averagely better than the baseline methods.

4.3. Results on the Image Aesthetics Dataset

The image aesthetic benchmark [18] consists of 10800 Flickr photos of four categories, i.e., "animals", "urban", "people" and "nature", and is constructed originally to retrieve beautiful yet unpopular images in social networks. The ground truths of the photos in the benchmark are five aesthetic grades: "Unacceptable" - images with extremely low quality, out of focus or underexposed, "Flawed" - images with some technical flaws and without any artistic value, "Ordinary" - standard quality images without technical flaws, "Professional" - professional-quality images with some artistic value, and "Exceptional" - very appealing images showing both outstanding professional quality and high artistic value. Fig. 4 shows an example from the "urban" category with one photo from each atheistic level. Each photo in the dataset is labeled by five graders of an online crowdsourcing platform to one of the five aesthetics levels. If the level of agreement is low, two more graders are recruited to perform the evaluation. In the experiments, these five aesthetic levels are indicated by rank 1 to 5, and the median rank of each image given by the graders is used as the ground truth. In each rank 75%, 5% and 20% images are randomly selected for training, validation and testing, respectively. All comparison methods are tested on five random training/testing partitions.

Table 4: Results on the image aesthetics dataset.

	Accuracy (%)					MAE				
	RED-SVM @8168	RED-SVM @deep	CNNm	Niu et al.	CNN-POR	RED-SVM @8168	RED-SVM @deep	CNNm	Niu et al.	CNN-POR
Nature	69.73	70.72	70.97	69.81	**71.86**	0.319	0.309	0.305	0.313	**0.294**
Animal	61.14	61.05	68.02	69.10	**69.32**	0.407	0.410	0.342	0.331	**0.322**
Urban	63.88	65.44	68.19	66.49	**69.09**	0.391	0.374	0.356	0.349	**0.325**
People	60.06	61.16	**71.63**	70.44	69.94	0.421	0.412	0.315	**0.312**	0.321
Overall	63.70	64.59	69.45	68.96	**70.05**	0.385	0.376	0.330	0.326	**0.316**

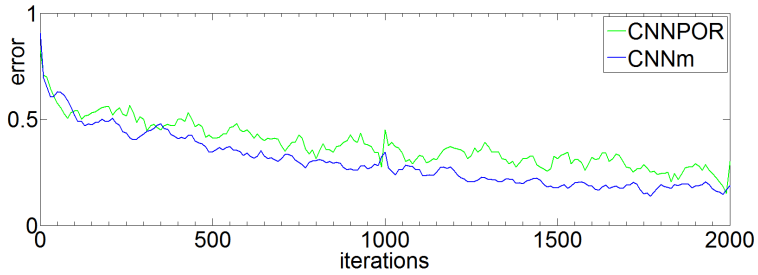

Figure 5: Training curves on the Adience face dataset.

Table 5: Results on the Adience face dataset.

Methods	Accuracy(%)	MAE
Feature-based [5]	45.1 ± 2.6	-
Lean DNN [11]	50.7 ± 5.1	-
CNNm	54.0 ± 6.3	0.61 ± 0.08
Niu et al.	56.7 ± 6.0	$\mathbf{0.54 \pm 0.08}$
CNNPOR	$\mathbf{57.4 \pm 5.8}$	0.55 ± 0.08

In the experiments, all the deep learning methods, including CNNm, Niu et al.'s method and CNNPOR, employ the VGG architecture and are fine-tuned from the ImageNet model. The images are resized to 256×256 pixels and are randomly cropped to 224×224 pixels further during the learning. The learning rate is set to 0.001 for the last fully-connect layer and 0.0001 for all other layers. RED-SVM is tested on the same 8168 features listed in Section 4.1 and the deep features extracted right before the first fully-connected layer. Table 4 summarizes the results in terms of accuracy and MAE. For both performance indexes, CNNPOR outperforms all the baseline methods on three categories. CNNm achieves the best performance for one category in terms of accuracy and Niu et al.'s method achieves the best performance for one category in terms of MAE.

4.4. Results on the Adience Face Dataset

To evaluate the scalability of CNNPOR, the Adience face dataset [11] is employed, which consists of 26580 Flickr photos of 2284 subjects and the ordinal ranks are eight age groups. In the experiments, the images alignment and five-fold partition follow [11]. Because the VGG net for multi-class classification has been verified scalable for large datasets, the training phase of CNNPOR is compared with CNNm and both methods are fine-tuned from the VGG ImageNet pretrained model. Same mini-batch size 96 is used for both CNNm and CNNPOR (i.e., $n = 32, m = 8, d = 8$ in Algorithm 1). Same learning rate 0.001 is applied for the last fully-connected layer of CNNm and G_c, G_r of CNNPOR, and 0.0001 for the rest layers. The C in Eq.5 is set to 1. Caffe package on Tesla M40 GPU is run for the experiments, and the average training time for one iteration of CNNm and CNNPOR is 3.3 and 3.6 seconds respectively.

Fig. 5 shows the training curves on one fold, which indicate the converge speed of CNNm and CNNPOR are similar, and in the experiments, both methods are trained for the same number of iterations 2000. Therefore, by employing the proposed efficient implementation, the scalability of CNNPOR is similar as CNNm. As shown in Fig. 5 and Table 5, the training error of CNNPOR is higher than CNNm, but CNNPOR achieves better performance on the testing set, which indicates the proposed method avoids overfitting effectively. RED-SVM is not scalable for this dataset, and the accuracy of state-of-the-art handcrafted feature-based method for this dataset is cited from [5] for comparison in Table 5. G. Levi and T. Hassner proposed a lean DNN [11] particularly for this dataset. They did not report MAE results in their papers. It is observed that CNNPOR achieves overall best performance consistently for all the benchmarks.

5. Conclusions

This paper proposes a new constrained optimization formulation for ordinal regression problems, and transforms it to an unconstrained optimization formulation with an effective deep learning implementation. The experimental results show that CNNPOR achieves overall the best results on all the four benchmarks, demonstrating the generality and scalability of the proposed method.

Acknowledgments

This work was conducted within Rolls-Royce@NTU Corporate Lab with support from the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

References

[1] J. S. Cardoso and J. F. Costa. Learning to classify ordinal data: The data replication method. *Journal of Machine Learning Research*, 8(Jul):1393–1429, 2007. 2, 5, 6

[2] J. Cheng, Z. Wang, and G. Pollastri. A neural network approach to ordinal regression. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1279–1284. IEEE, 2008. 1

[3] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. In *Journal of Machine Learning Research*, pages 1019–1041, 2005. 2

[4] W. Chu and S. S. Keerthi. Support vector ordinal regression. *Neural computation*, 19(3):792–815, 2007. 1, 2

[5] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014. 8

[6] E. Frank and M. Hall. *A simple approach to ordinal classification*. Springer, 2001. 2, 5, 6

[7] P. A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez. Ordinal regression methods: survey and experimental study. *Knowledge and Data Engineering, IEEE Transactions on*, 28(1):127–146, 2016. 2

[8] P. A. Gutiérrez, P. Tiňo, and C. Hervás-Martínez. Ordinal regression neural networks based on concentric hyperspheres. *Neural Networks*, 59:51–60, 2014. 1

[9] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002. 2, 3

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4, 7

[11] G. Levi and T. 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. 5, 8

[12] H.-T. Lin and L. Li. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367, 2012. 2, 4, 5, 6

[13] T.-Y. Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009. 2

[14] Y. Liu, X. Li, A. W. K. Kong, and C. K. Goh. Learning from small data: A pairwise approach for ordinal regression. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–6. IEEE, 2016. 4

[15] P. Martin, A. Doucet, and F. Jurie. Dating color images with ordinal classification. In *Proceedings of International Conference on Multimedia Retrieval*, page 447. ACM, 2014. 5, 6

[16] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. 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. 1, 2, 4, 5, 6

[17] F. Palermo, J. Hays, and A. A. Efros. Dating historical color images. In A. Fitzgibbon, S. Lazebnik, Y. Sato, and C. Schmid, editors, *ECCV (6)*, volume 7577 of *Lecture Notes in Computer Science*, pages 499–512. Springer, 2012. 5, 6

[18] R. Schifanella, M. Redi, and L. M. Aiello. An image is worth more than a thousand favorites: Surfacing the hidden beauty of flickr pictures. In *ICWSM*, pages 397–406, 2015. 5, 7

[19] A. Shashua and A. Levin. Taxonomy of large margin principle algorithms for ordinal regression problems. *Advances in neural information processing systems*, 15:937–944, 2002. 1

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4, 5, 6

[21] M. Wang, L. Yang, and X.-S. Hua. Msra-mm: Bridging research and industrial societies for multimedia information retrieval. *Microsoft Research Asia, Tech. Rep*, 2009. 5, 6