Implementation of Ordinal Regression with Multiple Output CNN for Age Estimation

Youcheng Li

[1]College of Artificial Intelligence, XJTU
West Xianning Road
Xi'an, Shaanxi, 710049,P.R. China
https://github.com/xjtulyc

Abstract

To address the non-stationary property of aging patterns, age estimation can be cast an ordinal regression problem. However, the process of extracting feature and learning a regression model are often separated and optimized independently in previous work. In Ordinal Regression with Multiple Output CNN for Age Estimation, the authors propose an End-to-End learning approach to address ordinal regression problems using deep Convolutional Neural Network, which could simultaneously conduct feature learning and regression modeling. In particular, an ordinal regression problem is transformed into a series of binary classification sub-problems, so that the collectively solve these tasks could be explored.

In this project, I implement the OR-CNN based on PyTorch and test it in Asian Face Age Dataset (AFAD) which contains more than 160K facial images with precise age ground-truths. The performance measured by the Mean Absolute Error (MAE) shows that ordinal regression model(Proposed OR-CNN) (MAE=3.34) is better than metric regression model(Proposed MR-CNN) (MAE=3.51).

Introduction

Estimation of human age from face images is still an active research topic with many applications such as demographic analysis, commercial user management, visual monitoring , and even aging progression. In previous methods, age estimation is often cast as a multi-class classification or metric regression problem. In a multi-class classification problem, the class labels are assumed to be independent to one another. However, the age labels have a strong ordinal relationship since they from a well-ordered set, which is not exploited in these multi-class classification methods.

On the other hand, metric regression approaches treat the age labels as numerical values to utilize such ordinal information for age estimation. However, the human face matures in different ways depending on the person's age. For example, facial aging effects appear as changes in the shape of face during childhood and changes in skin texture during adulthood. This property makes the random process formed by the aging patterns non-stationary in general. As is known to all, learning non-stationary kernels for a regression problem is usually different since it will easily cause over-fitting in the training process.

Due to the fact that facial aging process is a non-stationary process, one reliable information we can use would be their exact values. And hence the age estimation is cast a ordinal regression problem. For instance, Cao et al. formulated age estimation as a ranking problem and proposed a novel method based on Rank-SVM.

Recently, to directly utilize the well-studied classification algorithms, the ordinal regression problem is transformed into a series of simpler binary classification sub-problems. For each rank $k \in \{1, 2, ..., K-1\}$ a binary classifier is trained according to whether the rank of a simple larger than k. Then, the rank of a simple is predicted based on the classification results of the $K-1$ binary classifiers on this sample. A benefit of this kind of methods is that new generalization bounds for ordinal regression can be easily derived from known bounds for binary classification.

Inspired by it, the authors also transform ordinal regression as a series of binary classification sub-problems in this work. In particular, the Convolutional Neural Network is used to solved those binary classification sub-problems. Moreover, the CNNs has multiple output layers where each output layer corresponds to a binary classification task, called Multiple Output CNN in this work. Therefore, all the binary classifiers can be jointly trained in such a CNN. Since all the tasks share the same mid-level representations in the CNN, the correlation among distinct tasks could be explored, which is beneficial to improving the finial performance.

On the other hand, for either the metric regression based or the ordinal regression based approaches, the processes of extracting features and learning a regression model is separated and optimized independently. The most successful hand-crafted features for age estimation is the Bio-inspired Features (BIFs). Nevertheless, due to the unclear mechanism of how humane perceives the different aging pattern, it is still different to design good features for age estimation. On the contrary, in the approach we can conduct an End-to-End learning with CNN for age estimation, which could simultaneously optimize feature learning and regression modeling. As a result, we can automatically learn better features from facial images, and avoid directly designing hand-crafted features.

Briefly, the authors contributions are:

1. We propose to address ordinal regression problem using End-to-End deep learning methods.

2. We apply it to the task of age estimation, and achieve state-of-the-art results.

Ordinal Regression with CNN

Problem Formulation

Let us assume the i-th image is represented in an input $x_i \in X$, and there is an outcome space $y_i \in Y = \{r_1, r_2, ..., r_K\}$ with ordered ranks $r_K > r_{K-1} > ... > r_1$. The symbol $>$ denotes the ordering between different ranks. Given training samples $D = \{x_i, y_i\}_{i=1}^N$, the ordinal regression is to find a mapping from images to ranks $h(\cdot) : X \rightarrow Y$ such that – using a predefined cost $c : X \times Y \rightarrow R$ – the risk functional $R(h)$ is minimized.

In this paper, the cost matrix C is employed to measure the cost between predicted ranks and ground-truth ranks. In particular, C is a $K \times K$ matrix with $C_{y,r}$ being the cost of predicting an example (x, y) as rank r. Naturally, it is assumed that $C_{y,r} > 0$ for $r \neq y$. Particularly, the absolute cost matrix, which is defined by $C_{y,r} = |y - r|$, is a popular choice for general ordinal regression problems. Particularly, each age is often treated as a rank when applying ordinal regression algorithms to age estimation.

Approach

To directly utilize the well-studied classification algorithms, we transform ordinal regression as a series of binary classification sub-problem with K ranks is transformed into $K - 1$ simpler binary classification sub-problems. For each rank $r_k \in \{r_1, r_2, ..., r_{K-1}\}$, a binary classifier is constructed to predict whether the rank of a sample y_i is larger than r_k. And then the rank of an unseen sample is predicted based on the classification results of the $K - 1$ binary classifiers on this sample.

Specifically, the approach contains three steps: (a) given the original training data $D = \{x_i, y_i\}_{i=1}^N$, for the k-th binary classification sub-problem a specific training data is constructed as $D^k = \{x_i, y_i^k, w_i^k\}_{i=1}$, where the $y_i^k \in \{0, 1\}$ is a binary class label indicating whether the rank of the i-th sample y_i is larger than r_k as follows,

$$y_i^k = \begin{cases} 1 & if(y_i > r_k) \\ 0 & otherwise \end{cases} \quad (1)$$

And the w_i^k is the weight for the i-th example, i.e.,

$$w_i^k = |C_{y,k} - C_{y,k+1}| \quad (2)$$

Since absolute cost matrix is adopted in the approach, we have $\forall(i, k), w_i^k = 1$. (b) the $K - 1$ binary classifiers are trained with their corresponding training data. It is noticed that we adopt one CNN to collectively implement these binary classifiers in our approach. In particular, our CNN has a multiple-output structure where each output corresponds to a binary classifier. Thus, these binary classifiers are jointly trained in such a CNN; (c) the rank of an unseen sample x' is predicted as follows,

$$h(x') = r_q \quad (3)$$

$$q = 1 + \sum_{k=1}^{K-1} f_k(x') \quad (4)$$

where $f_k(x') \in \{0, 1\}$is the classification result of the k-th binary classifier of the sample x' (i.e., the k-th output of our multiple-output CNN). Ideally, these $f_k(x')$ should be consistent. However, ensuring the consistency is the training phase would significantly increase the training complexity. Hence we just apply Eq.3 without explicity ensuring the consistency among the different classifiers.

The benefits of the approach is two-fold: (1) an ordinal regression problem is solved by using an End-to-End deep learning method, so that we can automatically learn better features from facial images and avoid directly designing hand-crafted features. (2) the $K - 1$ classification subproblems are treated as $K - 1$ tasks, which are simultaneously solved with our multiple output CNN. Due to that all the tasks share the same mid-level representations in such a CNN, the correlation of distinct tasks could be explored, which is beneficail to improve the finial performance.

Architecture of the Multiple Output CNN

As shown in Fig.1, the network consists of 3 convolutional,4 local response normalization, and 2 max pooling layers followed by a fully connected layer with 80 neurons. At the input level, aligned face images of size $60 \times 60 \times 3$ are fed to the network as input. It is noted that color face images are used in this paper, which is different from gray images used in other papers. At the first convolutional layer, 20 kernels of size $5 \times 5 \times 3$ with stride of 1 pixels is applied on the input images. And after local response normalization and max-pooling operations, the feature maps of size $28 \times 28 \times 20$ are obtained.

The similar operations are conducted at the second and third convolutional layers with different kernel size. And then a fully connected layer with 80 neurons is used to generate a mid-level representation.

After that, the network branches out $K - 1$ output layers, where each output layer contains 2 neurons and corresponds to a binary classification task. The k-th task is to predict whether the age of the i-th facial image is larger than the rank r_k. For each task, the softmax normalized cross entropy loss is employed as loss function.

Learning the Multiple Output CNN

For a CNN with single output, we have N samples $\{x_i, y_i\}_{i=1}^N$, where x_i denotes the i-th image and y_i denotes the corresponding class label. For binary class label $y_i \in \{0, 1\}$, it is reasonable to employ cross-entropy as the loss function,

$$E_m = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} 1\{o_i^t = y_i^t\} w_i^t log(p(o_i^t | x_i, W^t)) \quad (5)$$

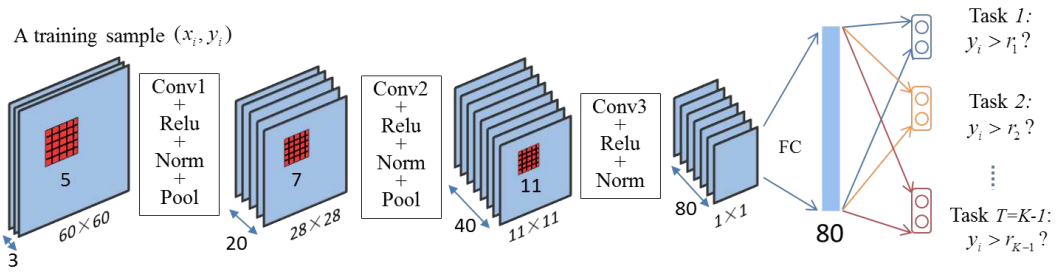

Figure 1: The architecture of the proposed Multiple Output CNN

where o_i indicates the output of the CNN for the i-th image, w_i indicates the weight of the i-th image, and W indicates the parameters of the entire CNN. Let W_l denotes the parameters of the l-th layer in the CNN, and hence we have $W = \{W_1, W_2, ..., W_{L-1}$. The Boolean test $1\{\cdot\}$ is 1 if the inner condition is true, and 0 otherwise.

For a CNN with $K-1$ outputs, each output corresponds to a distinct task. All the $T = K-1$ tasks (outputs) share the same N input images $\{x_i\}_{i=1}^N$, but have different class labels $\{\{y_i^t\}_{i=1}^N\}_{t=1}^T$. Let λ_t denotes the importance coefficient of the t-th task, the loss function of our multiple output CNN can be written as

$$E_m = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\lambda^t 1\{o_i^t = y_i^t\} w_i^t log(p(o_i^t|x_i, W^t)) \tag{6}$$

where o_i^t indicates the output of the t-th task for the i-th image, w_i^t indicates the weight of the i-th image fot the t-th task, and W^t indicates the parameters of the t-th task.

According to the architecture of out multiple output CNN, each task has a distinct output layer, but all the tasks share the same intermediate layers. Thus, only the parameters of the output layers for distinct tasks W_{L-1}^t are different from each other, i.e., we have $W_{L-1} = \{W_{L-1}^1, W_{L-1}^2, ..., W_{L-1}^T$. And the parameters of previous layers are the same as one another for all the tasks, i.e., $\forall l \in \{1, ..., (L-2)\}, W_l^1 = W_l^2 = ... = W_l^T = W_l$.

As shown in Fig.2, the learning procedure of the parameters for the output layer $W_{L-1} = \{W_{L-1}^t\}_{t=1}^T$ is similar to that of a single output CNN. For the t-th task, if the cross-entropy loss function is employed, the gradient of the weight from the j-the neuron in the layer $L-1$ to the k-th neuron in the layer L (i.e., $W_{L-1}^t(j,k)$) is computed as,

$$\frac{\partial E_m}{\partial W_{L-1}^t(j,k)} = \delta_L^t(k) o(j) \tag{7}$$

$$\delta_L^t(k) = p(o^t(k)|x_k, W_{L-1}^t) - 1\{o^t = y^t\} \tag{8}$$

where $o(j)$ is the output of the j-th neuron in the layer $L-1$, and $\delta_L^t(k)$ is the error of the k-th in the output layer.

The key is the learning procedure of the parameters for the penultimate layer. Since each neuron in the penultimate layer (i.e., layer $L-1$) is connected to all the neurons in the output layer (i.e., layer L), the error of the penultimate layer

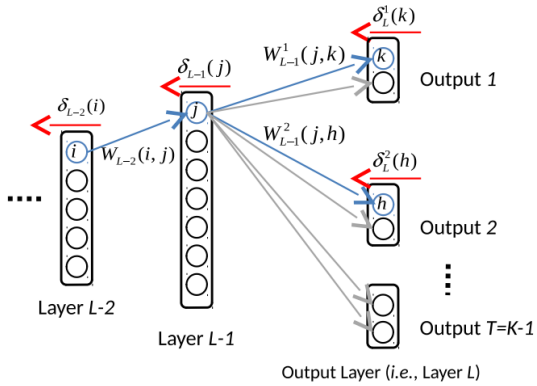

Figure 2: The back-propagation procedure for our Multiple Output CNN

δ_{L-1} is the integration of all the errors of output layers, as shown in Fig. 2. Specifically, the gradient of the weight from the i-th neuron in the layer $L-2$ to the j-th neuron in the layer $L-1$ (i.e., $W_{L-2}(i,j)$) is computed as,

$$\frac{\partial E_m}{\partial W_{L-2}(j,k)} = \delta_{L-1} t(k) o(i) \tag{9}$$

$$\delta_{L-1}(j) = \sum_{t=1}^{T}(\sum_{k \in L^t} \delta_L^t(k) W_{L-1}^t(j,k)) \tag{10}$$

where $o(i)$ is the output of i-th neuron in the layer $L-2$, and $\delta_{L-1}(j)$ is the error of the j-th neuron in the layer $L-1$. It is noticed that $\delta_{L-1}(j)$ is the weighted sum of the errors of output layers $\delta_L^t(k)$ over all the tasks, where the task-specific weights are $\lambda_t, (t = 1, ..., T)$.

The learning procedure of the weights for previous layers(i.e., $W_{L-3}, W_{L-4}, ...$) is the same as the standard learning procedure of CNN.

Experiment

Age Estimation

In this section, the performance of age estimation is compared among several methods including metric regression based and ordinal regression based method. For each

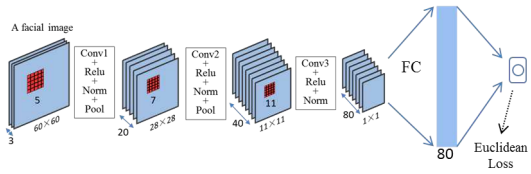

Figure 3: The architecture of network used in the method of Metric Regression with CNN (MR-CNN).

kind of methods, they can further categorized into two subcategories based on whether they used CNN learning algorithm.

The performance is measured by Mean Absolute Error(MAE) metric and the Cumulative Score(CS). The MAE is calculated using the average of the absolute errors between the estimated result and the ground truth. The cumulative score is calculated as follows:

$$CS(s) = -\frac{K_n}{K} \times 100\% \tag{11}$$

where K is the total number of test images, and K_n is the number of testing facial images whose absolute error between the estimated age and the ground truth age is not greater

We compared with metric regression based method and ordinal regression based methods. The performance is measured by the Mean Absolute Error(MAE) metric.

From Table 1, we have the following conclusions: (1) the ordinal regression based methods outperform the metric regression based methods in general; (2) more importantly, the integration of ordinal regression and deep learning methods could boost the performance significantly.

Comparing Metric and Ordinal Regression

Our approach presents an End-to-End CNN learning method, which transforms ordinal regression into a series of binary classification sub-problems. So it is still unclear which factor gives more contributions to the final improvement of performance. To take apart their distinctive contributions, we propose another baseline method, which only keeps the End-to-End CNN learning part and drops the part of transforming framework, i.e., it casts age estimation as a metric regression problem instead of an ordinal regression problem, and addresses the metric regression problem with an End-to-End CNN learning algorithm. For clarity, the baseline method is called Metric Regression with CNN (MR-CNN), while the proposed approach is called Ordinal Regression with CNN (OR-CNN).

For the MR-CNN, we need to solve a general metric regression problem with CNN, so we make a little modification on the architecture of our multiple-output CNN. Specifically, the fully connected layer is directly connected to one (the only one) output layer, and previous layers (i.e., three convolutional, local response normalization, max pooling layers) are kept. The output layer contains only one neurons, and its output indicates the regressed age for the input facial image. To train the MR-CNN, the L2-norm loss between ground-truth age and the regressed age is employed as the loss function. The architecture of the MR-CNN is shown in Fig.3.

Methods	AFAD
Proposed MR-CNN	4.07
Proposed OR-CNN	3.98

Table 1: The comparisons between metric regression and ordinal regression based methods. The performance is measured by the Mean Absolute Error (MAE) metric.

The comparison between MR-CNN and OR-CNN is conducted on both the AFAD datasets. From Table 1, we can find that the OR-CNN outperforms MR-CNN on the two datasets. In addition, we have done some statistical analysis for our OR-CNN model. We obtain MAE=4.07 and MAE=3.98 on the AFAD dataset respectively.

Those promising results illustrate that (1) it is necessary to cast age estimation as ordinal regression rather than a general metric regression problem.

Acknowledgments

Thank you for reading these instructions carefully.