Ordinal Regression using Noisy Pairwise Comparisons for Quetelet Index Range Estimation

Luisa F. Polanía   Dongning Wang   Glenn M. Fung
American Family Insurance, Strategic Data & Analytics, Madison, WI
{lpolania, dwang1, gfung}@amfam.com

ABSTRACT

Ordinal regression aims to classify instances into ordinal categories. In this paper, Quetelet index category estimation from facial images is cast as an ordinal regression problem. In particular, noisy binary search algorithms based on pairwise comparisons are employed to exploit the ordinal relationship among Quetelet index categories. Comparisons are performed with Siamese architectures, one of which uses the Bradley-Terry model probabilities as target. The Bradley-Terry model is a well-known approach to describe probabilities of the possible outcomes when elements of a set are repeatedly compared with one another in pairs. Experimental results show that our approach outperforms traditional classification-based methods at estimating Quetelet index categories.

KEYWORDS

Quetelet index, ordinal regression, Noisy binary Search, Siamese networks.

1 INTRODUCTION

Quetelet index (QI), also known as body mass index, is a biometric that provides important information about health condition and is frequently employed as a measure to diagnose obesity [21]. Diabetes, high blood pressure, high cholesterol, asthma and arthritis are frequently associated with overweight and obesity [18]. The traditional way to measure QI requires the presence of the subject and external elements, such as a measurement tape and a weight scale, which are not always available. Therefore, QI measurement from facial images is of interest for many applications where there is no access to monitored measurement devices. For example, health-related analysis using profile images from social media [12, 23], telemedicine kiosks to remotely diagnose patients [16] and face recognition [25]. However, QI estimation from face images is a challenging problem due to QI distribution variations across races and ages [12], and due to the lack of information since a body-dependent measure is attempted to be estimated with only facial data. Visual face and body information are not always correlated. For example, athletes typically have high QIs due to the high muscle mass in their bodies; however, their faces may resemble the face of a person with normal QI.

KDD'18, August 2018, London, UK

ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

Previous methods have been proposed for QI estimation using facial images [12, 13, 20, 24]. In [24], active shape models were employed for extraction of fiducial points, then seven features were built using face measurements based on those extracted points, *e.g.* face width to lower face height ratio. Finally, three regression methods, namely, support vector regression, Gaussian process, and least squares estimation, were used for the QI estimation. The authors used the MORPH-II database [20], which contains 55000 face images, to test their approach. Similarly, in [13], facial fiducial points were calculated for feature extraction, using a small dataset of 1124 face images. The problem was cast as a binary classification problem where the goal was to recognize normal QI population against overweight population with a Naive Bayes classifier. Convolutional neural networks (CNNs) have also been used for QI estimation. In [12], the VGG-Net model, which was pretrained on ImageNet, and the VGG-face model, which was pretrained with face images, was used for feature extraction. Epsilon support vector regression was used for QI estimation. The authors used a relatively small dataset consisting of 3368 images for training and 838 images for testing.

In this paper, the problem of QI category estimation is addressed. The same QI categorization proposed by the World Health Organization [8], as indicated in Table I, is used. A multi-class classification approach is not recommended for this problem, since it assumes independence between the class labels, which is not true for QI categories since they have a strong ordinal relationship. Instead, the QI category estimation problem is cast as an ordinal regression problem and addressed using a Noisy Binary Search (NBS) [10] approach, where the goal is to insert the QI associated to the test image into its proper place within the ordered sequence S, defined by the boundaries of the QI categories. That is, $S = \{16, 18.5, 25, 30, 35, 40\}$. In this paper, each sequence element from S is represented by a set of images, referred to as anchors, whose QI is very close or equal to the value of the sequence element.

Noisy binary search was first studied in the domain of channel coding with feedback [9]. Even though it was suggested in [10] that NBS algorithms could be employed for ranking problems, to the best of our knowledge, this is the first work that uses NBS for an ordinal regression application. Two algorithms previously proposed for NBS are employed in this paper, namely, Naive Noisy Binary Search (NNBS) [10] and Interval Noisy Binary Search (INBS) [6].

Noisy binary search relies on pairwise comparisons. A comparison operator that takes as input a given image and an anchor, and predicts if the QI of the given image is greater or smaller than the QI of the anchor with a small probability of error, is built to perform the pairwise comparisons needed by NBS. Two deep learning-based models are proposed to build the noisy comparator operator. Both models consist of Siamese-type architectures [4]. The differences between the architectures are related to the targets and the loss functions. One network uses the binary classes 0 and 1 as targets and the cross-entropy loss function. The other network uses the Bradley-Terry model probabilities [3] as target and the Euclidean loss function.

The proposed QI ordinal regression method is validated on the inmate active population dataset from the Florida Department of Corrections Database [1], which contains mugshots and QI information. Comparison results with classification approaches based on CNNs and handcrafted features suggest that the proposed approach achieves superior performance by exploiting the ordinal nature of the problem.

The contributions of this paper are summarized as follows:

- The application of NBS algorithms, which outperform classification methods based on CNNs and handcrafted features, to the problem of QI category estimation.

- Two Siamese-type architectures to calculate pairwise comparisons. The architectures include modifications with respect to the traditional Siamese architecture [12] used to learn similarity metrics. For example, both architectures incorporate the dot product between image feature vectors to further exploit correlation between the inputs. In addition, one of the architectures uses the Bradley-Terry model probabilities [3] as target.

- To the best of our knowledge, this is the first work that addresses the QI category estimation problem as an ordinal regression problem. Order in the QI categories is informative and exploiting it leads to improvements in performance.

- This work uses the largest dataset that has ever been used for the problem of QI estimation. The inmate active population dataset from the Florida Department of Corrections contains 96801 images, while the dataset used in [24], which is the largest dataset that has been used for QI estimation in the literature, contains 55000 images.

2 BACKGROUND

This section defines the QI biometric and the QI categories. It also describes the NBS problem and algorithms, the Siamese architecture typically used to learn similarity metrics, and the Bradley-Terry model.

2.1 Quetelet Index

Quetelet index is frequently employed as a measure to diagnose obesity [21]. It is calculated with weight and height information as follows

$$\mathrm{QI} = \frac{\text{Weight (kg)}}{(\text{Height (m)})^2}. \tag{1}$$

The World Health Organization proposed the QI categorization shown in Table 1 to classify subjects between underweight, normal, overweight, moderately obese, and severely obese [8].

2.2 Noisy Binary Search

The goal of NBS is the same goal of the traditional binary search problem of inserting an element x into its proper place within an ordered sequence $S = \{s_0, s_1, \ldots, s_{n-1}\}$ by comparing it with elements of the sequence. However, comparisons are noisy in NBS, *i.e.*, gives the wrong result with a small probability. Therefore, each element s_i of the sequence has an associated p_i corresponding to the probability that $s_i <= x$. The empirical probability $\hat{p}_i$, a proxy for p_i, is estimated by performing multiple comparisons between x and s_i.

Table 1: QI categorization proposed by the World Health Organization [8]

Category	QI range (kg/m^2)
Underweight	16-18.5
Normal	18.5-25
Overweight	25-30
Moderately obese	30-35
Severely obese	35-40

The work in [10] illustrates the NBS problem with the coin flip model in which each element from the sequence S is assigned a coin whose heads probability, p_i, is unknown. However, the head probabilities are assumed to be ordered $p_0 \geq p_1 \geq \ldots \geq p_{n-1}$ and we are allowed to toss the given coin to be able to estimate the empirical heads probability. If a coin is tossed $1/\epsilon^2$ times, then the probability that the estimated heads probability differs from p_i by more than ϵ is bounded by a constant from Chernoff bound [10]. The problem is solved when a pair of consecutive coins, i and $i+1$, such that the interval $[\hat{p}_i, \hat{p}_{i+1}]$ contains the number $1/2$, is found. In this paper, two algorithms to solve the NBS problem are considered, namely the NNBS and the INBS algorithms.

2.2.1 Naive Noisy Binary Search. The NNBS algorithm is recursive and resembles the traditional binary search algorithm. It maintains a set of indexes a and b which are initialized to s_0 and s_{n-1}, respectively. At each iteration, it tests the sequence element midway between a and b, denoted as c. If the calculated empirical probability associated to c, denoted as $\hat{p}_c$, is within $[1/2-\epsilon, 1/2+\epsilon]$, then the algorithm returns c. Otherwise, if $\hat{p}_c > 1/2+\epsilon$, then index a is updated with c and b remains unchanged. Similarly, if $\hat{p}_c < 1/2+\epsilon$, then index b is updated with c and a remains unchanged. The process repeats until either $\hat{p}_c \in [1/2-\epsilon, 1/2+\epsilon]$, in which case c is returned, or until a equals b, in which case the algorithm returns a. Details of the NNBS algorithm can be found in [10].

2.2.2 Interval Noisy Binary Search. Interval Noisy Binary Search modifies the NNBS algorithm by allowing backtracking [6]. It first builds a binary search tree of intervals such that the root node corresponds to sequence S. Each non-leaf node interval I has two children corresponding to the left and right halves of I. The leaves of the tree are the intervals between consecutive sequence elements. The algorithm starts at the root of the binary search tree and at every non-leaf node corresponding to interval I, it checks if the element to be searched, x, belongs to I by calculating the empirical probabilities associated to the sequence elements that define the boundaries of I. If either the empirical probability of the left boundary is smaller than 0.5 or the empirical probability of the right boundary is greater than 0.5, the algorithm backtracks to the current node's parent. Otherwise, if the test succeeds, and 0.5 lies within the empirical probability of the boundaries, the algorithm checks if x belongs to the left or right child by calculating the empirical probability associated with the middle element of I. If it is greater than 0.5, then it moves to the right child, otherwise, it moves to the left child. At a leaf node, the algorithm checks if x belongs to the corresponding leaf interval by maintaining a counter. The counter increases by one if 0.5 lies within the probability of the leaf interval boundaries. Otherwise, the counter decreases by one. If the counter becomes less than 0, the algorithm backtracks to the leaf's parent. The algorithm stops when the counter reaches a threshold K_1 and INBS returns the corresponding leaf node.

By following the above procedure, the algorithm may end up moving in a loop and never reaching the counter threshold K_1. If that is the case, the algorithm is run for a maximum of K_2 steps, saves all the visited sequence elements in a set Q, and runs NNBS on the set Q. Details of INBS can be found in [6].

2.3 Siamese architecture

Siamese convolutional neural networks have been widely employed to measure similarity in different applications, such as matching of image patches [26], face recognition [4] and signature verification [5]. All these problems fall into the category of matching problems where the goal is to detect if the inputs are similar or not. In [15], Siamese networks were used to rank images in terms of image quality.

There are twin branches in a Siamese network that share the same architecture and the same set of weights. Parameter training is mirrored across both branches.

Let z_i and z_j denote the feature representations of the inputs provided by the last layers of the twin branches. Siamese architectures are typically trained with the contrastive loss function as follows

$$\begin{aligned} L(\theta) &= \sum_{(z_i,z_j)\in D} y_{ij} d_{ij}^2 + (1-y_{ij})\max(0, m^2 - d_{ij}^2) \\ d_{ij} &= \|z_i - z_j\|_2, \end{aligned} \tag{2}$$

where D is the set of feature representation pairs produced by the last layers of the Siamese network across all the inputs, θ are the weights of the network, $y_{ij} \in \{0,1\}$ is the label with 1 and 0 denoting a matching and a non-matching pair, respectively. The first term of the loss function penalizes matching pairs whose feature representations are far apart while the second term penalizes non-matching pairs whose feature representations are closer than a margin m.

2.4 Bradley-Terry Model

The Bradley-Terry model is a probability model used to predict the outcome of a comparison [3]. Given a pair of individuals i and j drawn from some population, it estimates the probability that i beats j as

$$P(i \text{ beats } j) = \frac{\gamma_i}{\gamma_i + \gamma_j}, \tag{3}$$

where γ_i and γ_j are positive real-valued scores associated to individual i and j, respectively. For example, $P(i \text{ beats } j)$ may denote the probability that player i will win a game against player j and γ_i and γ_j may represent player strengths or abilities.

3 METHOD

This section describes the proposed method to address the problem of QI category estimation. Given an image x, the goal is to determine in which category from Table 1, the QI associated to image x belongs to. Addressing this problem with a classification approach, such as a traditional CNN, is not efficient since it disregards the ordinal nature of the problem. Instead, we propose to cast the problem as an ordinal regression problem and address it with an NBS approach.

3.1 Noisy binary search for QI ordinal regression

Let $S = \{16, 18.5, 25, 30, 35, 40\}$ be the sequence formed by the boundaries of the QI categories. Each element s_i of S is represented by a pool of images, referred to as anchors, such that their QI falls in the range $[s_i - \gamma, s_i + \gamma]$, where γ is a small constant, and therefore, the pool of images have QIs approximately equal to s_i. The goal is to insert the QI of x, denoted as x_{QI} into its proper place within S, by performing comparisons between x and the anchors in an NBS fashion. For this purpose, a comparison operator is built such that it outputs 1 if it predicts that the anchor image has a QI smaller or equal than x_{QI} and 0 otherwise. The comparison operator is noisy, *i.e.* it gives the wrong result with a small probability.

The proposed approach will be explained by using an analogy with the coin flip model presented in Section 2.2. In our problem, the equivalent of flipping the coin assigned to s_i is to randomly select an image from the anchors associated to s_i and run it with x through the comparison operator. The output of the operator, either 1 or 0, is the equivalent of the flip result, either head or tail. In Section 2.2, it was described that the coin assigned to s_i was flipped multiple times to calculate the empirical probability $\hat{p}_i$. Similarly, the comparison operator is run with several randomly selected anchor images assigned to s_i, to calculate the empirical probability $\hat{p}_i$ that $s_i \le x_{QI}$. Those probabilities are used by both the NNBS and the INBS algorithms to predict the right place for x_{QI} within the order sequence S, as explained in Section 2.2.

3.2 Comparison operator

This section presents comparison operators that are built with Siamese-type networks. The twin branches are truncated versions of traditional CNN architectures. Specifically, in this paper, the AgeNet architecture, which was proposed in [14] to predict age ranges from faces, and the VGG architecture, which was used in [12] for QI regression, are employed.

The AgeNet architecture is small and consists of three convolutional layers, each followed by Rectified Linear Unit (ReLU) and max-pooling layers, two fully connected layers, each followed by a ReLU and dropout layers, a third fully connected layer, which maps to the age class, and a softmax layer that assigns a probability for each class. The first, second, and third convolutional layers contain 96 filters of size $3 \times 7 \times 7$, 256 filters of size $96 \times 5 \times 5$, and 384 filters of size $256 \times 3 \times 3$, respectively. The first two fully connected layers contain 512 neurons each and the last layer contains 8 neurons corresponding to the age classes. Truncated versions of the AgeNet architecture, built by excluding the softmax and the last fully connected layer, are used as the twin branches of the Siamese network. The motivation for using a small and simple architecture, such as AgeNet, is that learning to compare input images with anchor images to predict which one has higher QI is intuitively easier than learning the nominal QI category.

The VGG architecture, which has more representational power than AgeNet, is also employed, at the expense of increasing the memory and computational cost. VGG contains 16 layers and all the convolutional layers have a receptive field of size 3×3. The convolution stride and the spatial padding are both fixed to 1 pixel. Each convolutional layer is followed by a ReLU layer. Spatial down-sampling is performed through max-pooling over a 2×2 pixel window, with stride 2, after 2 or 3 convolutional layers. A stack of 13 convolutional layers are followed by three fully-connected layers, where the first two have 4096 channels each, and the third has a number of channels that depends on the classification task. The last layer is the softmax layer. The activation function for each fully-connected layer is a ReLU as well. Another configuration for the twin branches of the Siamese network used in this paper corresponds to a truncated version of the VGG architecture, built by excluding the softmax and 2 last fully connected layers.

Feature outputs from the twin branches are concatenated. To further exploit the correlation of the features, the dot product between the features is included in the concatenation vector. The dot product is a modification with respect to the traditional Siamese architecture [12] used to learn similarity metrics, which typically only uses concatenation of features. In the case of the AgeNet-based Siamese network, two fully connected layers follow the concatenation. The first and second fully connected layers contain 512 and 1 neurons, respectively. In the case of the VGG-based Siamese network, three fully connected layers follow the concatenation. The first and second fully connected layers contain 2048 and 1024 neurons, respectively, and are followed by ReLU and dropout layers each. For both networks, the last fully connected layer contains a single output which is fed to a sigmoid function. The architecture of the Siamese network is illustrated in Fig. 1.

The comparison operator can be represented with the function $f(a_j^{s_i}, x; \theta)$, where θ denotes the network parameters and $a_j^{s_i}$ and x denote the inputs, which are the anchor image used for the jth comparison and the given image, respectively. The function takes the value 1 when the network predicts that the QI of $a_j^{s_i}$ is smaller or equal than the QI of x. Otherwise, it outputs 0. Let h_i denote the number of comparisons used to calculate the empirical probability $\hat{p}_i$ that $s_i \leq x_{QI}$, then the empirical probabilities that are fed to the NBS algorithms are defined by

$$\hat{p}_i = \frac{\sum_{j=0}^{h_i-1} f(a_j^{s_i}, x; \theta)}{h_i}, i = 0, \ldots, n-1. \tag{4}$$

3.2.1 Training modes. Two training modes for the Siamese network are used in this paper. Mode I uses the binary classes 1 and 0 as targets, where class 1 means that the QI of the anchor input image is smaller or equal than x_{QI} and class 0 means that the QI of the anchor input image is greater than x_{QI}. Mode I uses the cross-entropy loss function, which is defined as

$$L(\theta) = -\frac{1}{N_t} \sum_{i}^{N_t} [q_i \log g_i + (1 - q_i)\log(1 - g_i)], \tag{5}$$

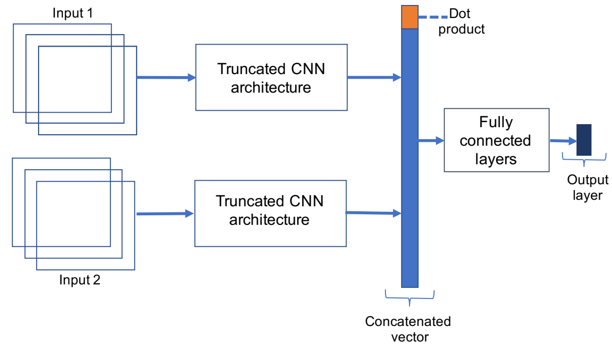

Figure 1: Schematic of the Siamese architecture

where N_t is the number of training image pairs, q_i is the truth label for the ith image pair and g_i is the corresponding predicted probability.

Mode II uses the Bradley-Terry model probabilities as target and the Euclidean loss function. The Bradley-Terry model is frequently used to model the outcome of games [3] and the motivation for using it in this paper is that comparison between a given image and the anchors associated to a sequence element have some similarity with the outcome of a game between a pair of players in the sense that the comparison operator predicts which subject from the input images wins at having higher QI. The randomness in the outcome of a game comes from the fact that the same player can perform differently at different times, while the randomness of the comparison operator at predicting if $s_i \leq x_{QI}$ comes from the fact that an anchor image is randomly selected from the pool of anchors at each comparison. The equivalent of the ability score γ_i associated to player i is the QI associated to the image. Therefore, the Bradley-Terry model probabilities, adapted to our problem, are defined by

$$P(x \text{ beats } a_j^{s_i}) = \frac{x_{QI}}{x_{QI} + s_i}, \quad i = 0, \ldots, n-1, \quad j = 0, \ldots, h_i - 1, \tag{6}$$

where, as before, $a_j^{s_i}$ denotes the randomly selected image from the pool of anchor images associated to s_i to perform the jth comparison.

The output probabilities of the mode II-trained network are mapped to the binary outputs of the comparison operator by using the criteria that if the probability is greater or equal than 0.5, then $f(a_j^{s_i}, x; \theta) = 1$. Otherwise, $f(a_j^{s_i}, x; \theta) = 0$.

3.2.2 Training of the Siamese networks. The procedure to build the training pairs for the networks is as follows. The entire dataset is first divided into training, denoted as training dataset I, validation and testing datasets. Images whose QI is within the range $[s_i - \gamma, s_i + \gamma]$ for each sequence element s_i are extracted from the training dataset I to build the anchor dataset. Let training dataset II denote the remaining set of training dataset I after extracting the anchors. A training budget b_i is assigned to each s_i and represents the number of training pairs assigned to s_i. A training pair is built by randomly selecting an image from the training dataset II and an anchor image. The budget b_i is equally distributed among the anchors belonging

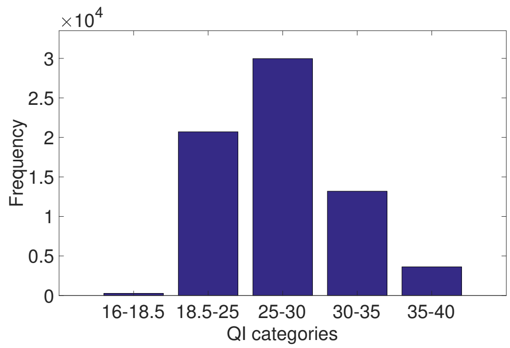

Figure 2: QI distribution of the training dataset

to s_i. The set of training pairs form the dataset used to train the Siamese networks. The same procedure is followed to build the validation pairs, but starting from the validation dataset.

For training the Siamese architectures, faces are first detected using the algorithm described in [17] and cropped to the size 224×224. The top branches of the AgeNet-based and VGG-based Siamese networks are initialized with the weights of the original AgeNet [14] and VGG-Face models [19], respectively. The fully connected layers following the feature concatenation are initialized with the Xavier method [7]. For the AgeNet-based network, the Adam optimizer [11] with a base learning rate of 1×10^{-4} and with default momentum values $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used for training with 300 samples per mini-batch. For the VGG-based network, the weights of the first 10 convolutional layers are kept frozen during training. Also, the Adam optimizer with a base learning rate of $1{\times}10^{-5}$ and with default momentum values $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used with 30 samples per mini-batch only due to memory constraints. The dropout probability is set to 0.5. Training stops when the loss on the validation pairs stops decreasing.

4 EXPERIMENTAL RESULTS

In this section, we present details of the database, ordinal regression quality metrics, and experiments used to evaluate the proposed method (code is available at https://github.com/lfpolani/QI_ordinal_regression).

4.1 Database

The Florida Department of Corrections is a downloadable database featuring records for all the inmates currently incarcerated in the Florida state prison system [1]. The database contains race, gender, weight and height information. The dataset is filtered to consider subjects with QI in the range 16-40, which corresponds to the categories of interest. Samples outside this range are rare, and in many cases, they correspond to noise in the data. For the experiments, 67756 randomly selected samples are chosen for training, 9045 for validation and 20000 for testing. The QI distribution of the training dataset across the different categories, shown in Fig. 2, indicates that this is an unbalanced dataset.

Figure 3: Image samples for each QI category

To illustrate the difficulty of the problem, Fig. 3 shows randomly selected images for each QI category. Note that distinguishing images from consecutive QI categories is visually challenging; especially distinguishing between normal and overweight images, which correspond to nearly 75% of the entire dataset. Therefore, learning pairwise comparisons is expected to also be challenging for the Siamese networks.

4.2 Performance metrics

Three metrics commonly used for ordinal regression problems [22] are used to evaluate the performance of the proposed method, namely accuracy (ACC), Mean Absolute Error (MAE), and the Kendall rank correlation coefficient (τ).

The accuracy is the fraction of correctly classified samples. Let $0, 1, \ldots, 4$ be the labels of the underweight, normal, overweight, moderately obese, and severely obese QI categories, respectively. Let y_i and $\tilde{y}_i$ denote the ground truth and predicted label for the testing sample x_i, respectively, and m denote the total number of samples in the testing set. Then, the MAE is defined by $\frac{1}{m}\sum_{i=0}^{m-1}|\tilde{y}_i - y_i|$. Given that 5 ordered labels are used, MAE values may range from 0 to 4.

The Kendall rank correlation coefficient is used to measure the correlation between the rankings of the ground truth and predicted labels. It is defined as

$$\tau = \frac{\sum_{0 \le i < j \le m-1}^{m-1} C((\tilde{y}_i, y_i), (\tilde{y}_j, y_j))}{m(m-1)/2}, \tag{7}$$

where $C(\cdot)$ is the concordance indicator function, defined by

Table 2: Performance of the AgeNet-based Siamese network

Sequence element	Mode I		Mode II	
	ACC	AUC	ACC	AUC
18.5	0.997	0.692	0.997	0.798
25	0.71	0.725	0.694	0.712
30	0.748	0.729	0.762	0.745
35	0.933	0.775	0.946	0.788

$$C((\tilde{y}_i, y_i), (\tilde{y}_j, y_j)) = \begin{cases} -1 & \text{if}(\tilde{y}_i - \tilde{y}_j)(y_i - y_j) < 0 \\ 1 & \text{if}(\tilde{y}_i - \tilde{y}_j)(y_i - y_j) > 0. \end{cases} \quad (8)$$

The Kendall rank correlation coefficient ranges from -1 to 1. For the perfect agreement and disagreement between the rankings of the ground truth and predicted labels, τ takes values 1 and -1, respectively. For completely independent rankings, τ has value 0.

For evaluating the performance of the Siamese networks, the accuracy and the commonly used area under the curve (AUC) are employed.

4.3 Performance of the Siamese architecture

This section evaluates the performance of the AgeNet-based Siamese network trained using modes I and II and using the validation pairs described in Section 3.2.2. The training budget per sequence element b_i is set to 150000. The value of γ needed to build the anchor set is set to 0.3. The classification threshold for the Siamese network trained using mode I is chosen such that the difference between the true positive rate and the false positive rate is maximized. This criteria leads to a classification threshold of 0.548. The classification threshold of the mode II-trained network is set to 0.5 to take advantage of the symmetry of the Bradley-Terry model around 0.5.

Table 2 shows the performance of the networks per sequence element. Note that sequence elements 16 and 40 are not included because the QI of the dataset always lies between those two numbers due to pre-filtering. The mode II-trained network outperforms the mode I-trained network for all the different sequence elements, except for 25. For mode I, the ground truth of the validation pairs associated to $s_i = 18.5$ are mostly 1 since most subjects have a QI greater than 18.5, and therefore, it is expected that the ACC associated to 18.5 should be high. Similarly, the ACC for $s_i = 35$ is expected to also be high since the validation pairs are highly unbalanced for that case as well. The AUC is known to be a better performance metric for unbalanced datasets [2] since it is defined as the the probability that a randomly chosen true positive will be ranked higher by the classifier than a randomly chosen true negative. Table 2 shows that the AUC attained by the Siamese networks mostly ranges from 0.7 to 0.8 across sequence elements.

4.4 Performance of the Noisy Binary Search algorithms

The performance of both NNBS and INBS is evaluated in this section using the testing dataset. First, the performance of the algorithms is analyzed as a function of the comparison budget. Then, the NBS algorithms are compared with the AgeNet and VGG classification networks and with a handcrafted feature-based method. For the NNBS algorithm, the value of ϵ is set 0.03. The part of the INBS algorithm that runs the NNBS on the sequence of visited elements Q also uses $\epsilon = 0.03$. The parameters K_1 and K_2 are set to $3 \log n$ and $12 \log n$, where n is the number of sequence elements.

4.4.1 Performance evaluation as a function of the comparison budget. A budget H is assigned to the NBS algorithm to perform pairwise comparisons. The fraction of the budget H assigned to a sequence element s_i, denoted as h_i, depends on the performance of the Siamese network at comparing with that particular sequence element. Thus, h_i is defined as

$$h_i = \frac{1 - \mathrm{AUC}_i}{\sum_{k=1}^{n-2}(1 - \mathrm{AUC}_k)}, \quad i = 1, \ldots, n-2 \quad (9)$$

where AUC_i is the AUC associated with the performance of the Siamese network at predicting if the QI of a given image is greater or equal than s_i (results shown in Table 2). The budget fraction is small for sequence elements for which the Siamese network performs well since less number of comparisons are expected to be needed for that case. Note that a budget is not assigned to s_0 and s_{n-1} because the QI of the testing samples always lies between s_0 and s_{n-1} due to pre-filtering of the data.

Figure 4 illustrates the performance of the NNBS algorithm by varying the budget H using the AgeNet-based Siamese network. The results in Figs. 4(a-c) correspond to the mean of the metrics across repetitions using different random anchor sampling at each repetition. Similarly, Figs. 4(d-f) correspond to the standard deviation of the metrics across repetitions. The number of repetitions used is set to 50. As expected, Figs. 4(a-c) show that the performance of the NNBS algorithm improves as the comparison budget increases. However, it improves very slowly for $H > 50$. The mode II-trained network outperforms the mode I-trained network for the entire comparison budget range, which suggests that using the Bradley-Terry model probabilities is better than using binary labels as target. As the comparison budget increases, the variance of the predicted empirical probabilities $\hat{p}_i$ across repetitions decreases, and therefore, the standard deviation of the performance metrics decreases as well.

Figure 5 compares the performance of the NNBS and the INBS algorithms as a function of H. Note that in the case of the INBS algorithm, the definition of the budget fraction in (9) applies at every step of the algorithm, but since backtracking may happen, the overall number of comparisons with s_i may exceed h_i. A smaller comparison budget implies more variance in the calculated empirical probabilities across comparisons, and therefore, for small values of H, it is is expected that the INBS algorithm backtracks to the parent node more often to retry comparisons and improve performance than for large values of H. This behavior reflects in Fig. 5(a), where the the INBS algorithm exhibits larger mean accuracy and smaller mean MAE than the NNBS algorithm for $H \leq 20$. Interval Noisy Binary Search also attains higher accuracy than NNBS for $H > 20$, but the difference is marginal for that case since there is less variance in the calculated empirical probabilities, and therefore, less backtracking. The MAE metric attained by INBS is smaller than that of NNBS for $H = 8$, but slightly bigger for $H > 20$. Regarding the τ metric, it achieves almost the same values for both algorithms when $H < 50$, but it is smaller for INBS than for NNBS

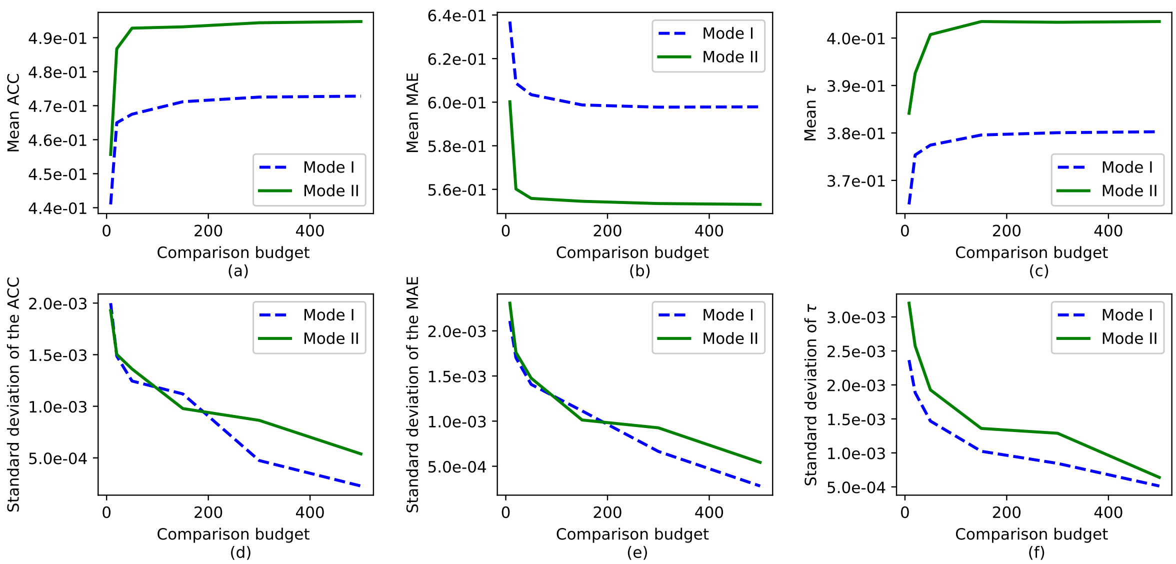

Figure 4: Performance of the NNBS algorithm as a function of the comparison budget H.

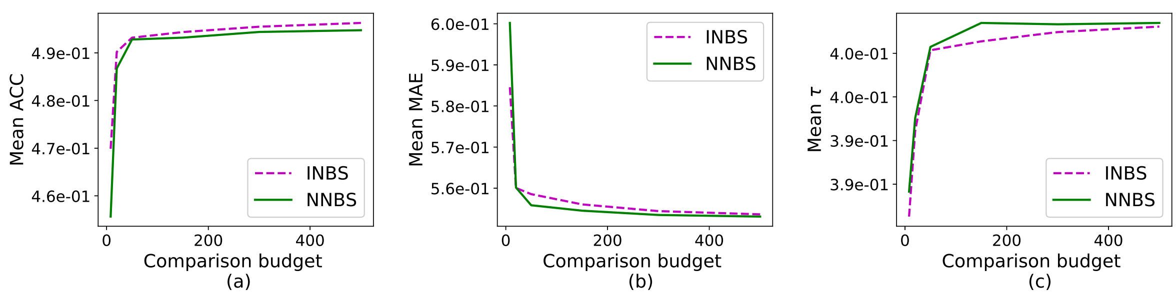

Figure 5: Comparison of the NNBS and INBS algorithms.

when $H \geq 50$. An explanation for the behavior of the MAE and τ metrics for $H \geq 50$ is that for a relatively large number of comparisons, less backtracking is expected and if the counter does not reach the threshold K_1, the INBS reduces to NNBS applied to only the visited elements, instead of the entire sequence S. From Fig. 5, we conclude that choosing INBS over NNBS is only worthy when a low comparison budget per iteration is required.

4.4.2 Increasing the representational power of the Siamese Network. An experiment to compare the performance of the NNBS algorithm using the AgeNet-based and VGG-based Siamese networks is presented in this section. The training mode I will be used for the experiment. The INBS is not included in this experiment because it was shown in Section 4.4 that it only outperforms NNBS when the comparison budget is low.

Table 3 compares the performance of the AgeNet-based and VGG-based Siamese networks at predicting if the QI of a given image is greater or equal than a sequence element, using the validation pairs described in Section 3.2.2. The results indicate that the VGG-based network outperforms the AgeNet-based network, which is expected given that VGG has more representational power than AgeNet. Only for the sequence element 25, the ACC obtained with

Table 3: Comparison between the AgeNet-based and VGG-based Siamese networks

Sequence element	AgeNet-based		VGG-based	
	ACC	AUC	ACC	AUC
18.5	0.997	0.692	0.997	0.743
25	0.71	0.725	0.7	0.739
30	0.748	0.729	0.772	0.758
35	0.933	0.775	0.945	0.822

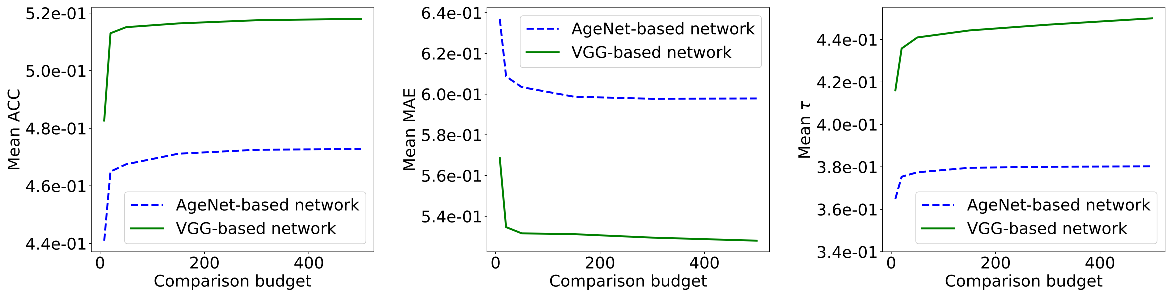

Figure 6: Performance of the NNBS algorithm using the AgeNet-based and VGG-based Siamese networks

the VGG-based network is slightly smaller than the ACC of the AgeNet-based network.

Fig. 6 compares the performance of the NNBS algorithm using the AgeNet-based and VGG-based Siamese networks. As expected, using the VGG-based network in the NNBS algorithm leads to better results for the entire comparison budget range. For both networks, the performance improves slowly when the comparison budget exceeds 50.

4.4.3 Performance comparison with classification-based approaches. In this section, the proposed method is compared with the AgeNet and VGG classification networks and with a handcrafted feature-based method.

The AgeNet and VGG classification networks employed in this experiment use the original AgeNet and VGG architectures presented in Section 3.2, respectively, with the exception of the last fully connected layer which uses 5 outputs to match the number of QI categories. The training procedure for the networks is similar to that described in Section 3.2.2. Faces are first detected and cropped to the size of 224×224. The AgeNet and VGG networks are initialized with the weights of the original AgeNet model [14] and VGG-Face model [19], respectively, except for the last fully connected layer of the networks, which is initialized with the Xavier method [7]. The same optimizer, base learning rate, batch size, dropout factor and training stopping criteria of the AgeNet-based and VGG-based Siamese networks are employed for the AgeNet and VGG classification networks, respectively. The VGG architecture was previously used in [12] for QI regression.

The handcrafted feature-based method uses the same geometric features proposed in [24] for QI regression. The features are based on the extraction of fiducial facial points. A multiclass linear-kernel SVM is employed for the QI range estimation. The penalty parameter C of the error term is estimated using grid search, which leads to $C = 1$.

Table 4 compares the performance of the NNBS and INBS algorithms using a budget $H = 500$ with the performance of the AgeNet and VGG classification networks and with the handcrafted feature-based method. The value of $H = 500$ is selected for the comparison since it corresponds to the best performance achieved by the NBS algorithms. Note that the metrics in Table 4 for the NBS algorithms correspond to the mean values across repetitions using different random anchor sampling at each repetition. Results show that NBS algorithms outperform classification-based methods by exploiting the ordinal nature of the QI range estimation problem. The NBS algorithms using the AgeNet-based Siamese network outperform the results of the AgeNet classification network. Similarly, the NNBS algorithm using the VGG-based Siamese network outperforms the results of the VGG classification network. Note that the τ metric results indicate that the rankings of the ground truth are much more correlated with the rankings of the labels predicted by the proposed methods than with the rankings of the labels predicted by the handcrafted feature-based method.

Table 4: Comparison of the NBS algorithms with classification-based methods

Method	ACC	MAE	τ
Handcrafted feature-based method	0.45	0.615	0.13
AgeNet	0.467	0.616	0.362
NNBS (AgeNet-based, Mode I)	0.473	0.598	0.38
NNBS (AgeNet-based, Mode II)	0.495	0.553	0.403
INBS (AgeNet-based, Mode II)	0.496	0.554	0.403
VGG	0.494	0.555	0.45
NNBS (VGG-based, Mode I)	0.518	0.528	0.45

5 CONCLUSIONS

Noisy binary search has been studied extensively in the area of computer science. In this paper, it was shown how the ability of NBS algorithms to exploit the ordinal nature of a problem can be leveraged to address the problem of QI range estimation. As NBS relies on pairwise comparisons, two Siamese networks were proposed to perform the comparisons. The motivation for using a method based on pairwise comparisons was that predicting which subject from two images has higher QI is intuitively easier than learning the nominal QI category. Experimental results show that the proposed method leads to superior performance when compared with classification-based methods that do not exploit order in the data. Even though the proposed method is applied to QI range estimation in this paper, it can be extended to other ordinal regression applications.

REFERENCES

[1] Florida department of corrections. http://www.dc.state..us/ActiveInmates/search.asp. Florida Department of Corrections database of inmate photos [Online database].

[2] U. Bhowan, M. Johnston, and M. Zhang. Developing new fitness functions in genetic programming for classification with unbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):406–421, 2012.

[3] R. Bradley and M. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546. IEEE, 2005.

[5] S. Dey et al. Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017.

[6] M. Falahatgar, A. Orlitsky, V. Pichapati, and A. Suresh. Maximum selection and ranking under noisy comparisons. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1088–1096, International Convention Centre, Sydney, Australia, 06–11 Aug 2017.

[7] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[8] W. Health Organization et al. Global database on body mass index: an interactive surveillance tool for monitoring nutrition transition. *Public Health Nutr*, 9(5):658–60, 2006.

[9] M. Horstein. Sequential transmission using noiseless feedback. *IEEE Transactions on Information Theory*, 9(3):136–143, 1963.

[10] R. Karp and R. Kleinberg. Noisy binary search and its applications. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 881–890. Society for Industrial and Applied Mathematics, 2007.

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] E. Kocabey et al. Face-to-bmi: Using computer vision to infer body mass index on social media. *arXiv preprint arXiv:1703.03156*, 2017.

[13] B. Lee, J. Jang, and J. Kim. Prediction of body mass index from facial features of females and males. *International Journal of Bio-Science and Bio-Technology*, 4(3):45–62, 2012.

[14] 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.

[15] X. Liu, J. van de Weijer, and A. D. Bagdanov. Rankiqa: Learning from rankings for no-reference image quality assessment. *CoRR*, abs/1707.08347, 2017.

[16] C. Lowe and D. Cummin. The use of kiosk technology in general practice. *Journal of telemedicine and telecare*, 16(4):201–203, 2010.

[17] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European Conference on Computer Vision*, pages 720–735. Springer, 2014.

[18] A. Mokdad, E. Ford, B. Bowman, W. Dietz, F. Vinicor, V. Bales, and J. Marks. Prevalence of obesity, diabetes, and obesity-related health risk factors, 2001. *Jama*, 289(1):76–79, 2003.

[19] O. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[20] K. Ricanek and T. Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *International Conference on Automatic Face and Gesture Recognition*, pages 341–345. IEEE, 2006.

[21] A. Romero-Corral et al. Accuracy of body mass index in diagnosing obesity in the adult general population. *International journal of obesity*, 32(6):959–966, 2008.

[22] J. Sánchez-Monedero, P. Gutiérrez, P. Tiňo, and C. Hervás-Martínez. Exploitation of pairwise class distances for ordinal classification. *Neural computation*, 25(9):2450–2485, 2013.

[23] I. Weber and Y. Mejova. Crowdsourcing health labels: Inferring body weight from profile pictures. In *Proceedings of the 6th International Conference on Digital Health Conference*, pages 105–109. ACM, 2016.

[24] L. Wen and G. Guo. A computational approach to body mass index prediction from face images. *Image and Vision Computing*, 31(5):392–400, 2013.

[25] L. Wen, G. Guo, and X. Li. A study on the influence of body weight changes on face recognition. In *IEEE International Joint Conference on Biometrics*, pages 1–6. IEEE, 2014.

[26] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.