Deep Ordinal Classification with Inequality Constraints

Soufiane Belharbi1  Ismail Ben Ayed1  Luke McCaffrey2  Eric Granger1
1LIVIA, École de technologie supérieure, Université du Québec, Montreal, Canada
2Rosalind and Morris Goodman Cancer Research Centre, Dept. of Oncology, McGill University, Montreal, Canada
soufiane.belharbi.1@ens.etsmtl.ca, {ismail.benayed,eric.granger}@etsmtl.ca
luke.mccaffrey@mcgill.ca

Abstract

This study investigates a new constrained-optimization formulation for deep ordinal classification. We impose unimodality of the label distribution implicitly via a set of inequality constraints over pairs of adjacent labels. To tackle the ensuing challenging optimization problem, we solve a sequence of unconstrained losses based on a powerful extension of the log-barrier method. This accommodates standard SGD for deep networks, and avoids computationally expensive Lagrangian dual steps and projections, while outperforming substantially penalty methods. Our non-parametric model is more flexible than the existing deep ordinal classification techniques: it does not restrict the learned representation to a specific parametric model, allowing the training to explore larger spaces of solutions and removing the need for ad hoc choices, while outperforming substantially penalty methods. Our non-parametric model is more flexible than the existing deep ordinal classification techniques: it does not restrict the learned representation to a specific parametric model, allowing the training to explore larger spaces of solutions and removing the need for ad hoc choices, while scaling up to large numbers of labels. It can be used in conjunction with any standard classification loss and any deep architecture. We also propose a new performance metric for ordinal classification, as a proxy to measure a distribution uni-modality, referred to as the Sides Order Index (SOI). We report comprehensive evaluations and comparisons to state-of-the-art methods on benchmark public datasets for several ordinal classification tasks, showing the merits of our approach in terms of label consistency and scalability. A public reproducible PyTorch implementation is provided.

1. Introduction

Standard classification losses for training deep neural networks, e.g., cross-entropy, do not impose any prior on the structure of the labels. Designed to penalize the error between predicted and true labels for each data sample, they do not account for the semantic relationships that might exist between the labels. However, in a wide range of classification tasks, the set of labels exhibits a natural structure, for instance, in the form of a specific order. A typical example is classifying biopsy samples, with the labels encoding cancer-aggressiveness levels (or grades), which are ordered. Ordinal classification (OC) attempts to leverage such natural order of the labels, and is useful in a breadth of applications, such as movies rating [11, 28], market bonds rating [32], age estimation [29, 35, 44], emotion estimation [24, 42, 43], cancer grading [18], diabetic retinopathy grading [3], photographs dating [34], among many others.

Fig. 1 depicts a typical OC task, which consists of grading cancer (8 classes, for illustration). It shows two different posterior distributions (PDs) of class predictions for the same input sample with the same probability of the predicted label g5 and the same cross-entropy loss: d1 corresponds to paradoxical predictions, ranking grade g1 right after predicted grade g5, despite their significant semantic difference, whereas d2 corresponds to consistent (ordered) predictions.
We report comprehensive evaluations and comparisons to As an example, consider d when important actions are associated with the predictions. Therefore, in general, they require several task-dependent This is often done by enforcing uni-modality either on the la-
immediate-surgery-with-intensive-chemotherapy

numbers of labels. Furthermore, we introduce a new perfor-
ad hoc
training to explore larger solution spaces and removing the representation to a specific parametric model, allowing the
penalty methods. Our non-parametric model is more flexible than the existing techniques – it does not restricts the learned

modal prior, e.g., Poisson [3] or Gaussian [15, 16, 17, 23]. Therefore, in general, they require several task-dependent choices, including tuning carefully the hyper-parameters that control the form (or shape) of the parametric model and complex network-architecture design. They might also lead to models that do not scale well for large numbers of labels, as is the case of [3]

Contributions: We propose a novel constrained-optimization formulation of ordinal classification. Our model enforces uni-modality and label-order consistency via a set of inequality constraints over pairs of adjacent labels, which can be imposed on any standard classification loss and integrated with any deep architecture. To tackle the ensuing challenging optimization problem, we solve a sequence of unconstrained losses based on a powerful extension of the log-barrier method, well-known in the context of interior-point methods [7, 27]. This accommodates standard SGD, and avoids computationally expensive Lagrangian dual steps and projections, while outperforming substantially standard penalty methods. Our non-parametric model is more flexible than the existing techniques – it does not restricts the learned representation to a specific parametric model, allowing the training to explore larger solution spaces and removing the need for ad hoc choices, while allowing to scale up to large numbers of labels. Furthermore, we introduce a new performance metric for OC, as a proxy to measure a distribution uni-modality, referred to as the Sides Order Index (SOI). We report comprehensive evaluations and comparisons to state-of-the-art methods on benchmark public datasets for several OC tasks (breast cancer grading, age estimation, and historical color image dating). The results indicate that our approach outperforms state-of-the-art methods in terms of consistency, while scaling up to large numbers of labels.

2. Related Work

Several recent work converted the hard target label into a prior distribution [15, 16, 17, 23]. One way to impose such a prior distribution over the labels is, for instance, to optimize a divergence loss, such as Kullback–Leibler (KL), for training the network [16]. A typical choice in the literature is to use a parametric uni-modal Gaussian to model label distribution, with the mean of the Gaussian encoding the true label, while the variance is set through validation or through prior knowledge. The main motivation behind this class of label-distribution methods is to deal with the ambiguity and uncertainty of discrete labels, in tasks such as age estimation, head pose estimation, and semantic segmentation. Using prior distributions over the labels is also related to the well-known label-smoothing regularization for improving the training of deep networks [40]. Such a regularization perturbs the hard label with a uniform distribution, embedding uncertainty in the ground truth. In [10], the prior order of the labels is encoded using a step function. Instead of a standard one-hot encoding, a binary-vector encoding the labels is used as a target for network training. Another class of methods in literature penalize directly the deviations of the softmax predictions of the network from a uni-modal Gaussian, which is constrained to have the same mean as the true label and minimal variance [2, 35]. These methods do not impose specific prior knowledge on the variance, but attempt to push it towards zero. In [3, 12] specific parametric distributions are directly encoded within the network output, e.g., binomial or Poisson distributions. The network output is a single scalar used to model the output distribution and to infer the probability of each label.

While the above mentioned techniques enforce the prior order and uni-modality of the output distributions, they have several shortcomings. The techniques in [15, 16, 17, 23] constrain the output distribution to have a specific form (or shape), following the choice of a parametric uni-modal model, e.g., Gaussian. This requires ad hoc (manual) setting of model parameters, e.g., the variance of a Gaussian, which might have a direct but unclear impact on the results. The models in [2, 35] learn such variance parameter by pushing it towards zero, yielding a sharp Gaussian that approaches a Dirac function. Choosing a sharp or flat Gaussian has a direct impact on the labels and their order, but it is not clear how to make such a choice. In general, the choices that one has to make as to the form of the parametric model are task-dependent. Directly encoding specific parametric distributions within the network output, as in [3], also requires several choices, including complex network architecture design and setting ad hoc parameters. In particular, the
performance of the method in [3] seems to depend strongly on a hyper-parameter that controls the variance of the distribution. Such a hyper-parameter should be set empirically with some care since its value changes the distribution shape from uniform to Gaussian. The method in [3] was evaluated on only two ordinal datasets that have very few classes (5 and 8). The authors claim that the method does not scale well to large numbers of labels, in particular, with the Poisson distribution, due to the nature of the latter. With this distribution choice, the label probabilities have a stair-like shape with a constant (deterministic) variation in each step.

As detailed in the next section, our proposed approach circumvents the need to pre-define a parametric uni-modal model for network outputs, and to set its parameters. We enforce uni-modality and a consistent order between the labels, but without constraining the learned representation to any specific parametric model, allowing the training to explore a larger space of solutions. To this end, we describe the uni-modality property through ordering adjacent labels, thereby ensuring decreasing monotonicity of the probabilities on both sides of the target label. Such an order is represented using a set of inequality constraints that are optimized with a powerful extension of the log-barrier method [7, 27], well-known in the context of interior-point methods and convex optimization. Unlike [3], our method scales up to a very large number of labels. Furthermore, we provide a new metric, as a proxy, to assess the uni-modality of a distribution by measuring the order between adjacent labels.

3. Uni-modality and order-consistency via inequality constraints

Let us consider a set of training samples \( D = \{(X^{(i)}, y^{(i)})\}_{i=1}^{n} \) where \( X^{(i)} \) is an input sample, a realization of the discrete random variable \( X \) with support set \( \mathcal{X} \); \( y^{(i)} \) is the sample label, a realization of the discrete random variable \( y \) with support set \( \mathcal{Y} = \{1, \ldots, c\} \) that exhibits an overall order between the labels, \( y_0 \prec y_1 \prec \ldots \prec y_{c-2} \prec y_{c-1} \), where \( a \prec b \) means that the event described by the label \( a \) is ordered before\(^4\) the event described by the label \( b \). In this work, we aim at enforcing the uni-modality of the posterior probability, with the latter decreasing monotonically as we move further away from the target label. For notation simplicity and clarity in this section, we drop the sample index \( (\cdot)^{(i)} \), and the expected value of losses over all the samples. We define \( \hat{p}(y|X) \) as the posterior probability estimated by a neural-based model \( \mathcal{M}(X; \theta) \), \( s \in \mathbb{R}^n \) denotes the logit scores\(^5\) obtained by the model \( \mathcal{M} \), where the posterior probability is computed using standard softmax function, \( \hat{p}(k|X) = \exp(s_k) / \sum_{j=1}^{n} \exp(s_j) \). We note dom \( f \) and ran \( f \) the domain and range of the function \( f \), respectively.

We describe the uni-modality of a function with respect to a target point (or label) as a decreasing monotonicity of the function above and below the point, according to some pre-defined order. To ensure such decreasing monotonicity, it suffices to embed hard constraints on the order between every two adjacent points, within each of two sets of points, one including those below the target and the other including those above. Instead of ordering probabilities, we consider ordering scores. For a sample \( (X, y) \), and its score vector \( s \), we formulate adjacent ordering as a constrained-optimization problem using the following set of hard inequality constraints:

\[
\begin{align*}
\text{minimize} & \quad C(\mathcal{M}(X; \theta), y) \\
\text{subject to} & \quad s_k < s_{k+1}, \text{ for } k < y, \quad \text{ and } \\
& \quad s_{k+1} < s_k, \text{ for } y \leq k < c, \\
\end{align*}
\]

where \( C(\cdot, \cdot) \) is a classification loss function such as cross-entropy (CE). This loss will be considered in our experiments, but our constrained optimization can be used in conjunction with any loss function.

\[
C(\mathcal{M}(X; \theta), y) = -\log \hat{p}(y). 
\]

Our constrained optimization problem in Eq.(1) is very challenging for modern deep networks involving large numbers of trainable parameters [26, 27, 31, 37, 38]. In the context of deep networks, hard constraints are typically addressed with basic penalty methods [21, 25, 26] as they accommodate SGD optimization, avoiding explicit primal-dual steps and projections. However, for a large set of constraints, penalty methods might have difficulty guaranteeing constraint satisfaction as they require careful and manual tuning of the weight of each constraint. In principle, standard Lagrangian-dual optimization seeks automatically the optimal weight of the constraints, and have well-established advantages over penalty methods, in the general context of convex optimization [7]. However, as shown and discussed in several recent works [31, 37, 38, 26, 27], in contexts other than ordinal classification, those advantages do not materialize in practice for deep networks due mainly to the interplay between the dual steps and SGD optimization, causing instability, and to the incurred computational complexity. To solve our problem in Eq.(1), we consider two alternatives to explicit Lagrangian-dual optimization: a penalty-based method and a powerful extension of the log-barrier method,\(^6\)

\(^3\)For the Poisson distribution, one can show that the difference between the probabilities of two adjacent labels within the same PD is constant and depends only the single score. The difference between any two labels is constant as well, and depends on the score and the labels separating them.

\(^4\)The term ‘before’ here refers to any type of progressive ordering, with respect to a fixed set of labels, induced by any type of factor: Δtime (e.g. age), Δspace (e.g. distance), disease aggressiveness (e.g. cancer grading), ...

\(^5\)For generality, we assume that such scores are unbounded.
which well-known in the context of interior-point methods [7, 27]. In particular, the log-barrier method is well-suited to our problem. It approximates Lagrangian optimization via implicit dual variables, handling effectively large numbers of constraints, while accommodating standard SGD and avoiding explicit Lagrangian-dual steps and projections.

3.1. Penalty-based optimization

We augment the loss $C(\cdot, \cdot)$ by converting each inequality constraint into a penalty. In the general context of optimization [6], PN-based methods approximate a constrained optimization problem with an unconstrained one by adding a penalty term $H$, which increases when the corresponding constraint is violated. By definition, a penalty $H$ is a non-negative, continuous, and differentiable function that has zero value only when the constraint is satisfied [6]. In this work, we consider the following quadratic penalty for imposing inequality constraint $a < b$:

$$H(\delta_a^b) = \begin{cases} (\delta_a^b - \epsilon)^2 & \text{if } \delta_a^b \geq 0 , \\ 0 & \text{otherwise ,} \end{cases}$$

where $\delta_a^b = a - b$, $a, b \in \mathbb{R}$, and $\epsilon \in \mathbb{R}^+_\epsilon$ is a slack constant to avoid the equality case. In this case, our problem in Eq.(1) becomes:

$$\min_{\theta} C(\mathcal{M}(X ; \theta), y) + \lambda \left( \sum_{y=1}^{y-1} H(\Delta_{k+1}(s)) + \sum_{j=y}^{c-1} H(\Delta_{k+1}(s)) \right),$$

where $\Delta_{k}(s) = s_m - s_l$, and $\lambda \in \mathbb{R}^+$ is a penalty coefficient that indicates the importance of the augmented term; and which is determined empirically. For later comparison, we refer to this penalty method as PN (Eq.4).

3.2. Log-barrier-based optimization

While PN-based methods are simple and straightforward, they do not guarantee constraints satisfaction and require an empirical tuning of the importance coefficient(s) [14, 19]. Moreover, once a constraint is satisfied, the penalty is zero. Consequently, constraints that are satisfied in one iteration may not be satisfied in the next one since the penalty does not play a role of a barrier at the feasible set of solutions. This can be problematic when dealing with a large number of constraints at once. To avoid such well known issues with PN methods, we consider log-barrier methods (LB) as an alternative. LB-methods belong to interior-point methods (IP) [7], which aim to approximate Lagrangian optimization with a sequence of unconstrained problems and implicit dual variable, avoiding dual steps and projections [7]. Eq.(1) can be re-written in a standard form of a LB method:

$$\min_{\theta} C(\mathcal{M}(X ; \theta), y)$$

subject to $\Delta_{k+1}(s) < 0$, for $k < y$, and

$$\Delta_{k+1}(s) < 0, \text{ for } y \leq k < c.$$  

LB methods are widely used for inequality constrained problems [7]. The main aim of LB methods is to convert a constrained problem of the form in (5) into an unconstrained one via an indicator function $I(\cdot)$ that has zero penalty when the constraint is satisfied, and a penalty of $+\infty$ otherwise. Instead of using $I$, LB methods employ an approximate, $\hat{I}$, using the logarithmic function, where the penalty decreases the further we get away from violating the inequality, forming a barrier between feasible and infeasible solutions. To solve Eq.(5), a strictly feasible set of parameters $\theta$ are required as a starting point. Such a set is found through Lagrangian minimization of inequality constraints (phase I [7]), which turns out to be a problem of similar difficulty as the constrained optimization in Eq.(5) [7]. Such strategy makes standard LB methods impractical for constraining deep models. We use an extension of the standard log-barrier based on a different approximation of the indicator function [27]. The main advantage is that this algorithm does not require starting from a feasible solution – i.e., $\text{dom} \ I$ is no longer restricted to feasible points of $\theta$. A direct consequence is that stochastic optimization techniques such as SGD can be directly applied without the need for a feasible starting point. We replace problem Eq.(5) by the following sequence of unconstrained problems, parameterized by a temperature $t$:

$$\min_{\theta} C(\mathcal{M}(X ; \theta), y) + \left( \sum_{j=1}^{y-1} \hat{I}(\Delta_{k+1}(s)) + \sum_{j=y}^{c-1} \hat{I}(\Delta_{k+1}(s)) \right),$$

where $\hat{I}$ is a log-barrier extension, which is convex, continuous, and twice differentiable [27]:

$$\hat{I}(r) = \begin{cases} -\frac{r}{t} \log(-r) & \text{if } r \leq -\frac{1}{t} \\ tr - \frac{1}{t} \log(-r) + \frac{1}{t} & \text{otherwise .} \end{cases}$$

One can show that optimizing log-barrier extensions approximate Lagrangian optimization of the original constrained problem with implicit dual variables, with a duality gap upper bounded by a factor of $1/t$; see Proposition 2 in [27]. Therefore, in practice, we use a varying parameter $t$ and optimize a sequence of losses of the form (6), increasing gradually the value of $t$ by a constant factor. The network parameters evaluated at the current $t$ and epoch are used as a starting point for the next $t$ and epoch. For later comparison, we refer to this LB method as the extended log-barrier (ELB) (Eq.(6)).
4. Experiments

4.1. Performance metrics:

We denote $\hat{y}$ as the label predicted by the model, and $T$ as an evaluation dataset. For our evaluation, we report performances using three metrics:

- The classification accuracy: $\text{ACC} = \frac{1}{|T|} \sum_{(x,y) \in T} \mathbb{1}[\hat{y} = y]$;

- The mean absolute error, which is often used in OC setups: $\text{MAE} = \frac{1}{|T|} \sum_{(x,y) \in T} |\hat{y} - y|$; and

- We propose a new metric that measures how well labels are ordered, above and below the true or predicted label. Following the *non-monotonic index* defined in [5, 20], we propose the Side-Order Index (SOI) that counts the number of satisfactions (non-violations) of the order constraints over adjacent labels above and below a reference label $\nu$,

$$
\text{SOI}_\nu = \frac{1}{|T|} \sum_{(x,y) \in T} \left[ \frac{1}{c-1} \sum_{j=1}^{\nu-1} \mathbb{1}[\Delta_k^{j+1}(\bar{\nu}) < 0] + \sum_{j=\nu}^{c-1} \mathbb{1}[\Delta_k^{j+1}(\bar{\nu}) < 0] \right]
$$

(8)

This metric can be seen as a proxy to describe how well a distribution is uni-modal with respect to a reference label $\nu$. Furthermore, it is appropriate for evaluating the performance of a constrained-optimization method for our problem in (1) as it evaluates constraint satisfaction. We compute the expected value over a normalized measure so that the metric is independent of the total number of labels, hence, independent of the number of pairs of $c-1$ adjacent labels. Consequently, the range of the measure is in $[0, 1]$, 0 indicating that all the adjacent labels are un-ordered, and 1 indicating a perfect order.

In our experiments, we consider two cases of the reference label $\nu$:

- **The predicted label (SOI$_\nu$):** In this case, we measure the consistency of the model’s predictions with respect to the predicted label and *independently* from the true label. This measure is the most important as it assesses the model’s consistency when it is evaluated in a real scenario, where the true label is unknown.

- **The true label (SOI$_\nu$):** In this case, we measure the consistency of the model’s predictions with respect to the true label. This measure will be noisy when the model predicts the wrong label. To obtain a good performance with this measure, the model must predict the correct label *and* yield a correct order of the labels. Therefore, this measure integrates the performance of both classification and label-order consistency. Also, it provides a common ground to compare different methods as the reference label is unique.

4.2. Datasets and training protocol:

We consider datasets that naturally exhibit order between the labels. We target datasets that have large number of labels. Applying OC over datasets with very small number of labels is unlikely to show the power and limitation of different methods. We consider 3 different applications: breast cancer grading, photographs dating, and age estimation using public benchmarks.

(1) **Breast cancer grading:** This task consists in classifying histology images of breast biopsy into different grades that are ordered with respect to the cancer aggressivity. We use the dataset BACH (Part A) Breast cancer [1] referred to here as HCI, along with the experimental protocol in [34]. The dataset contains a total of 400 images, and 4 classes: normal, benign, in Situ, and invasive (in this order). Following the protocol described in [4, 39], we perform two splits of the dataset where in each split we take 50% of samples per class for test, and perform 5-fold cross-validation to build the train/validation sets.

(2) **Photographs dating:** This task consists of predicting in which decade a color image was taken. We consider the dataset Historical Color Image dataset (for classification by decade)\(^6\) [34], referred to here as HCI, along with the experimental protocol in [34]. The dataset contains 5 decades from 1930s to 1970s, each containing 265 images, for a total of 1325 images. We took 50 random images per decade to form a test set, with the rest of the data used for training and validation sets using 10-fold cross-validation. This process is repeated 10 times.

(3) **Age estimation:** This task consists of estimating someone’s age based on their picture. We consider three datasets:

- FG-NET dataset\(^7\) [36], referred to here as FGNET. It is a very early database used for age estimation, which contains 1,002 face images from 82 individuals, with ages ranging from 0 to 69 (70 classes);

- Asian Face Age Dataset (AFAD)\(^8\) [33] light (AFAD-Lite), with a total of 59,344 samples, an age range from 15 to 39, and a total number of classes of 22; and

- AFAD-Full [33], with a total of 165,501 samples, an age range from 15 to 72, and a total number of classes of 58.

---

\(^6\) [http://graphics.cs.cmu.edu/projects/historicalColor/](http://graphics.cs.cmu.edu/projects/historicalColor/)

\(^7\) [https://yanweifu.github.io/FG_NET_data/index.html](https://yanweifu.github.io/FG_NET_data/index.html)

\(^8\) [http://afad-dataset.github.io/](http://afad-dataset.github.io/)
The same protocol is conducted over the three datasets. Following the experimental setup in [8, 9, 33, 41], we randomly select 20% of the entire dataset for testing, and perform 5-fold cross-validation to build train and validation sets. This process is repeated 10 times.

Dividing samples into train, validation, and test sets is done in the following way. Samples of the same individual do not end up in different sets. Without any additional information about the individuals, the partition of the samples within each set is done in a way that preserves balanced proportions over the labels. When additional information about the individuals is available, e.g., gender, we first partition the data with respect to this information, then with respect to the labels, eventually merging everything so as to maintain balanced proportions across all the three sets. All the splits are done randomly using a deterministic code that we provide publicly along with the splits.

5. Results and Discussion

The quantitative results obtained with the different methods and over the different test sets are presented in Tabs.1-

Tab.3. Fig.2 depicts some results for qualitative appraisal. We recall the notation of the different methods – CE (Eq.2): cross-entropy method; REN [10]: re-encode the hard target into a vector of binary values and use the mean squared error as a loss. The threshold is set to 0.5; LD [15, 16, 17, 23]: label distribution learning with Bayes rule prediction. The variance is set to 1; MV [35]: mean-variance loss combined with softmax where the predicted label is the round function of the expected value of the labels. Following [35], we set \(\lambda_1 = 0.2, \lambda_2 = 0.05\); PO [3]: Hard-wire Poisson distribution at the network output, and use cross-entropy for learning. \(\tau\) is fixed and set to 1 as in the paper [3], and the prediction is based on the expected value of the labels; PN (Eq.4): penalty-based method; ELB (Eq.6): extended log-barrier method. All the methods have the same exact model capacity (i.e., ResNet18 [22]), except PO [3] where we add a dense layer that maps from \(c\) (number of classes) into one (positive score). Based on the obtained results, we note the following.

1) In term of SOI\(\hat{y}\) consistency, we observe that ELB method yields the best results, achieving almost perfect ordering SOI\(\hat{y}\) > 97% over all datasets. Then, comes second the PN method, but with a large gap of 55% < SOI\(\hat{y}\) < 89% in comparison to ELB, and a slightly better performance than CE (53% < SOI\(\hat{y}\) < 81%). We note that the gap between PN and ELB increases with the number of classes: \(\sim 15%\), \(\sim 20\%\) on ICIAR and HCI with 4 and 5 classes, respectively; \(\sim 26\%\), \(\sim 31\%\) on AFAD-Lite with 22 classes, and AFAD-Full with 58 classes, respectively; and \(\sim 42\%\) on FGNET with 70 classes. This is expected, since the PN method does not cope well with the interplay between different constraints, more so when the number of constraints is large unlike LB methods which approximate Lagrangian optimization.

The different methods REN, LD, MV, and PO yield a SOI\(\hat{y}\) not far from the CE method. LD achieves good results over ICIAR and HCI with SOI\(\hat{y}\) of \(\sim 91\%\) and \(\sim 84\%\) ranking in the second place after ELB. However, its performance drops to \(\sim 66\%\), \(\sim 61\%\) over AFAD-Lite -Full, and to \(\sim 51\%\) over FGNET suggesting that it does not handle manage well large number of classes. Compared to LD, MV performs better on large number of classes achieving \(\sim 83\%\), \(\sim 70\%\), \(\sim 58\%\) over AFAD-Lite -Full, and FGNET, respectively. REN performance is usually worst than CE. The case of PO is particular. It is expected to obtain SOI\(\hat{y}\) = 100% but since the predicted label is the expected value of the label (and not the \(\text{argmax}\) of the scores), it achieves a low level of performance. When PO uses \(\text{argmax}\), it does not represent a fair comparison to other methods since the order in PO is hardwired, while the order is learned with all other methods. PO [3] does not scale up to large numbers of classes, and, in [3], it was evaluated only on two datasets, with 5 and 8 classes.
Table 1. Evaluation of different methods over the test sets of ICIAR and HCI datasets (classification datasets).

| Method   | ICIAR      | HCI        |
|----------|------------|------------|
|          | ACC (%)    | MAE (%)    | SOI_y (%) | SOI_y (%) | ACC (%) | MAE (%) | SOI_y (%) | SOI_y (%) |
| CE (Eq.2)    | 84.47 ± 1.85 | 0.19 ± 0.02 | 75.27 ± 2.07 | 80.05 ± 1.68 | 56.44 ± 2.28 | 0.68 ± 0.05 | 67.62 ± 1.80 | 78.05 ± 1.67 |
| REN [10]          | 79.99 ± 2.79 | 0.23 ± 0.028 | 54.77 ± 1.33 | 54.94 ± 1.38 | 52.27 ± 2.43 | 0.63 ± 0.03 | 54.85 ± 1.47 | 57.08 ± 1.46 |
| LD [16]          | 62.91 ± 1.66 | 0.49 ± 0.03 | 77.53 ± 1.01 | 91.02 ± 1.33 | 37.95 ± 2.36 | 1.00 ± 0.04 | 66.67 ± 1.33 | 84.42 ± 1.33 |
| MV [35]          | 43.01 ± 3.23 | 0.61 ± 0.04 | 75.89 ± 2.45 | 62.99 ± 2.09 | 50.18 ± 2.06 | 0.69 ± 0.03 | 71.20 ± 1.74 | 71.82 ± 1.93 |
| PO [3]           | 43.21 ± 2.61 | 0.60 ± 0.02 | 66.39 ± 0.71 | 61.27 ± 1.63 | 46.96 ± 2.08 | 0.71 ± 0.03 | 52.17 ± 1.00 | 49.37 ± 1.69 |
| PN (Eq.4)        | 84.42 ± 1.40 | 0.19 ± 0.02 | 79.81 ± 1.74 | 84.82 ± 1.40 | 56.67 ± 2.30 | 0.50 ± 0.05 | 60.68 ± 1.87 | 80.48 ± 1.64 |
| ELB (Eq.6)       | 83.21 ± 3.79 | 0.19 ± 0.04 | 93.13 ± 1.52 | 99.21 ± 0.41 | 54.90 ± 2.40 | 0.65 ± 0.04 | 83.86 ± 1.22 | 99.21 ± 0.32 |

Table 2. Evaluation of different methods over the test sets of AFAD-Lite and AFAD-Full datasets (regression datasets). -- indicates that the method does not scale to a large number of classes.

| Method   | AFAD-Lite | AFAD-Full |
|----------|-----------|-----------|
|          | MAE (%)    | SOI_y (%) | SOI_y (%) | MAE (%)    | SOI_y (%) | SOI_y (%) |
| CE (Eq.2)    | 3.69 ± 0.06 | 61.76 ± 0.81 | 68.67 ± 1.00 | 3.73 ± 0.06 | 61.03 ± 0.47 | 63.69 ± 0.47 |
| REN [10]          | 3.00 ± 0.01 | 57.89 ± 0.41 | 58.91 ± 1.16 | 3.19 ± 0.02 | 55.25 ± 0.43 | 55.97 ± 0.30 |
| LD [16]          | 5.03 ± 0.02 | 57.11 ± 12.0 | 66.52 ± 0.19 | 5.15 ± 0.028 | 57.03 ± 0.08 | 61.82 ± 0.12 |
| MV [35]          | 2.96 ± 0.02 | 76.90 ± 0.79 | 83.32 ± 0.95 | 3.20 ± 0.02 | 68.08 ± 0.55 | 70.56 ± 0.58 |
| PO [3]           | 3.56 ± 0.02 | 26.48 ± 0.03 | 20.34 ± 0.41 | ---         | ---         | ---         |
| PN (Eq.4)        | 3.67 ± 0.06 | 65.00 ± 0.62 | 72.64 ± 0.75 | 3.73 ± 0.21 | 64.03 ± 0.52 | 67.00 ± 0.60 |
| ELB (Eq.6)       | 3.08 ± 0.03 | 85.10 ± 0.28 | 98.83 ± 0.60 | 3.49 ± 0.06 | 93.45 ± 0.15 | 98.47 ± 0.29 |

Figure 2. Each sub-figure is the overlapped posterior PD of all the test samples of FGNET dataset (split 0, fold 0) that have been predicted with the same label (35) using a model trained with the following losses (from left to right): CE, PN, and ELB. The x-axis is the label $y$ support set $Y$, and the y-axis is the posterior PD $p(y|X)$. The PDs are overlapped with an opacity equals to 1/number of overlapped distributions. The darker the blue color, the more frequent the probability. (Best visualized in color.)

$SOI_y$ behaves similarly, but with less magnitude since $SOI_{\hat{y}}$ is its upper bound. These results show the amount of disorder with a CE prediction in an OC setup, which confirms its inadequacy to such contexts. Furthermore, we observe the power of LB methods compared to PN method, with the former achieving much better constraint satisfaction than the latter. This shows the effectiveness of LB methods and, in particular, its extension [27], for optimizing with inequality constraints on model outputs (Fig.3).

(2) **In term of MAE**, and over datasets with very few labels, we do not observe major differences between the different methods. Over FGNET, all the methods obtain a slightly higher error compared to the recently reported state-of-the-art result of $MAE = 3.17$ in [36]. Over AFAD-Lite, the method yields $MAE = 2.96$, while REN obtains a state-of-the-art error over AFAD-Full with $MAE = 3.19$, compared to value $MAE = 3.34$ reported in [33]. All our experiments are repeated 10 times, while [33] repeats experiments for 100 times. Over HCI, ELB and REN yielded $MAE = 3.19$.

On the FGNET dataset, which has 70 labels, ELB, PN, and CE obtained $MAE = 5.14, 7.76, 7.86$, respectively. Such a performance is in the same range as the results reported in [36], with the best performance of $MAE = 3.17^{10}$. We note that REN obtains state-of-the-art results over AFAD-Full with an $MAE = 3.16$, compared to [33] with $MAE = 3.34$. However, we repeat our experiments 10 times, while [33] repeat for 100 times.

(3) **In term of ACC**, we observe that the CE, PN, and ELB
methods obtain relatively similar results over ICIAR and in the same range as in [39]. Similar behavior is observed over HCI, with PN ahead, achieving $\text{ACC} = 56.67\%$. [30] reports $\text{ACC} = 50.23\%$. In the OC literature, $\text{ACC}$ is not reported since the aim is to get as close as possible to the target. Therefore, $\text{ACC}$ can be high, and has less importance.

(4) Qualitative results: The aforementioned quantitative results show the benefits of our proposal in term of reinforcing the posterior consistency, independently of the number of labels. We find that the visual results over datasets with very few labels do not inform much about the uni-modality issue. The obtained results over AFAD-Lite, AFAD-Full and FGNET, which have large numbers of labels provide us great insights. Fig. 2 depicts the overlapped posterior probabilities of all the test samples predicted with the same label over FGNET. The figure shows clearly that the probability obtained by CE is not uni-modal. Similarly, PN does not really improve much compared to CE. However, we note a major improvement and the clear uni-modality in the prediction of LB methods, confirming again the benefits of such optimization methods. This uni-modality is obtained while surpassing the need to pre-define any distribution, nor dealing with any of its parameters, which is the aim of this work.

(5) Training time: PN and LB methods do not add a significant computation overhead compared to CE.

6. Conclusion

We presented a new constrained-optimization formulation for ordinal classification, with uni-modality of the label distribution imposed implicitly via a set of inequality constraints over pairs of adjacent labels. To tackle the ensuing challenging optimization problem, we solve a sequence of unconstrained losses based on a powerful extension of the log-barrier method, which is well-known in the context of interior-point methods. This accommodates standard SGD, and avoids computationally expensive Lagrangian dual steps and projections, while outperforming substantially standard penalty methods. Our non-parametric model is more flexible than the existing ordinal classification techniques: it does not restrict the learned representation to a specific parametric model, allowing the training to explore larger spaces of solutions and removing the need for ad hoc choices, while scaling up to large numbers of labels. It can be used in conjunction with any standard classification loss and any deep architecture. We also propose a new performance metric for ordinal classification, as a proxy to measure a distribution uni-modality, referred to as the Sides Order Index (SOI). We report comprehensive evaluations and comparisons to state-of-the-art methods on benchmark public datasets for several ordinal classification tasks, showing the merits of our approach in terms of label consistency and scalability.

References

[1] G. Aresta, T. Araújo, S. Kwok, et al. Bach: Grand challenge on breast cancer histology images. *medical image analysis*, 56:122 – 139, 2019.
[2] C. Beckham and C. Pal. A simple squared-error reformulation for ordinal classification. CoRR, abs/1612.00775, 2016.

[3] C. Beckham and C. Pal. Unimodal probability distributions for deep ordinal classification. In ICML, volume 70, pages 411–419, 2017.

[4] S. Belharbi, J. Rony, J. Dolz, I. Ben Ayed, L. McCaffrey, and E. Granger. Min-max entropy for weakly supervised pointwise localization. CoRR, abs/1907.12934, 2019.

[5] A. Ben-David. Monotonicity maintenance in information-theoretic machine learning algorithms. Machine Learning, 19:29–43, 1995.

[6] D.P. Bertsekas. Athena scientific. Nonlinear programming, 1995.

[7] S. Boyd and L. Vandenberghe. Convex optimization. Cambridge university press, 2004.

[8] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In CVPR, pages 585–592, 2011.

[9] K. Chen, S. Gong, T. Xiang, and C. Change Loy. Cumulative attribute space for age and crowd density estimation. In CVPR, pages 2467–2474, 2013.

[10] J. Cheng, Z. Wang, and G. Pollastri. A neural network approach to ordinal regression. In IEEE Int. Joint Con. on Neural Networks, pages 1279–1284, 2008.

[11] K. Crammer and Y. Singer. Pranking with ranking. In NIPS, pages 641–647, 2001.

[12] J. P. Da Costa and J. S. Cardoso. Classification of ordinal data using neural networks. In ECML, pages 690–697, 2005.

[13] T. Durand, T. Mordan, N. Thome, and M. Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In CVPR, volume 2, pages 5957–5966, 2017.

[14] Roger Fletcher. Practical methods of optimization john wiley & sons. New York, 80, 1987.

[15] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, and X. Geng. Deep label distribution learning with label ambiguity. IEEE Transactions on Image Processing, 26:2825–2838, 2017.

[16] X. Geng. Label distribution learning. IEEE Transactions on Knowledge and Data Engineering, 28:1734–1748, 2016.

[17] X. Geng, C. Yin, and Z.-H. Zhou. Facial age estimation by learning from label distributions. Transactions on Pattern Analysis and Machine Intelligence, 35:2401–2412, 2013.

[18] A. E. Gentry, C.K. Jackson-Cook, D.E. Lyon, and K.J. Archer. Penalized ordinal regression methods for predicting stage of cancer in high-dimensional covariate spaces. volume 14, pages CIN–S17277, 2015.

[19] Philip E Gill, Walter Murray, and Margaret H Wright. Practical optimization. London: Academic Press, 1981.

[20] P.A. Gutiérrez and S. García. Current prospects on ordinal and monotonic classification. Progress in Artificial Intelligence, 5:171–179, 2016.

[21] F.S. He, Y. Liu, A.G. Schwing, and J. Peng. Learning to play in a day: Faster deep reinforcement learning by optimality tightening. In ICLR, 2017.

[22] K. He, X. Zhang, S.g Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016.

[23] Z. Huo, X. Yang, C. Xing, Y. Zhou, P. Hou, J. Lv, and X. Geng. Deep age distribution learning for apparent age estimation. In CVPR workshops, pages 17–24, 2016.

[24] X. Jia, X. Zheng, W. Li, C. Zhang, and Z. Li. Facial emotion distribution learning by exploiting low-rank label correlations locally. In CVPR, pages 9841–9850, 2019.

[25] Z. Jia, X. Huang, I. Eric, C. Chang, and Y. Xu. Constrained deep weak supervision for histopathology image segmentation. IEEE trans. on medical imaging, 36:2376–2388, 2017.

[26] H. Kervadec, J. Dolz, M. Tang, E. Granger, Y. Boykov, and I. Ben Ayed. Constrained-cnn losses for weakly supervised segmentation. medical image analysis, 54:88–99, 2019.

[27] H. Kervadec, J. Dolz, J. Yuan, C. Desrosiers, E. Granger, and I. Ben Ayed. Constrained deep networks: Lagrangian optimization via log-barrier extensions. CoRR, abs/1904.04205, 2019.

[28] Y. Koren and J. Sill. Ordrec: An ordinal model for predicting personalized item rating distributions. In ACM Conf. on Recommender Systems, pages 117–124, 2011.

[29] H. Liu, J. Lu, J. Feng, and J. Zhou. Ordinal deep feature learning for facial age estimation. In IEEE Int. Con. on Automatic Face & Gesture Recognition, pages 157–164, 2017.

[30] Y. Liu. Ordinal regression based on data relationship. PhD thesis, 2019.

[31] P. Márquez-Neila, M. Salzmann, and P. Fua. Imposing hard constraints on deep networks: Promises and limitations. In CVPR Workshop on Negative Results in Computer Vision, 2017.

[32] J. Moody and J. Utans. Architecture selection strategies for neural networks: Application to corporate bond rating prediction. In Neural networks in the capital markets, pages 277–300, 1994.

[33] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Ordinal regression with multiple output cnn for age estimation. In CVPR, pages 4920–4928, 2016.

[34] F. Palermo, J. Hays, and A.A. Efros. Dating historical color images. In ECCV, pages 499–512, 2012.

[35] H. Pan, H. Han, S. Shan, and X. Chen. Mean-variance loss for deep age estimation from a face. In CVPR, pages 5285–5294, 2018.

[36] G. Panis, A. Lanitis, N. Tsapatsoulis, and T.F. Cootes. Impose constraints on deep networks: Promises and limitations. In CoRR, abs/1904.04205, 2019.

[37] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In ICCV, pages 1796–1804, 2015.

[38] S.N. Ravi, T. Dinh, V. Lokhande, and V. Singh. Explicitly imposing constraints in deep networks via conditional gradients. In ICLR, pages 585–592, 2019.

[39] J. Rony, S. Belharbi, J. Dolz, I. Ben Ayed, L. McCaffrey, and E. Granger. Deep weakly-supervised learning methods for classification and localization in histology images: a survey. CoRR, abs/1909.03354, 2019.
[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In CVPR, pages 2818–2826, 2016. 2

[41] X. Wang, R. Guo, and C. Kambhamettu. Deeply-learned feature for age estimation. In WACV, pages 534–541, 2015. 6

[42] H. Xiong, H. Liu, B. Zhong, and Y. Fu. Structured and sparse annotations for image emotion distribution learning. volume 33, pages 363–370, 2019. 1

[43] Y. Zhou, H. Xue, and X. Geng. Emotion distribution recognition from facial expressions. In annual con. on multimedia, pages 1247–1250, 2015. 1

[44] H. Zhu, Y. Zhang, G. Li, J. Zhang, and H. Shan. Ordinal distribution regression for gait-based age estimation. CoRR, abs/1905.11005, 2019. 1