Ordinal Quantification through Regularization

Mirko Bunse[1][0000−0002−5515−6278] (✉), Alejandro Moreo[2][0000−0002−0377−1025], Fabrizio Sebastiani[2][0000−0003−4221−6427], and Martin Senz[1][0000−0002−9377−3939]

[1] Department of Computer Science
TU Dortmund University
44227 Dortmund, Germany
{mirko.bunse,martin.senz}@cs.tu-dortmund.de
[2] Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
{alejandro.moreo,fabrizio.sebastiani}@isti.cnr.it

Abstract. Quantification, i.e., the task of training predictors of the class prevalence values in sets of unlabelled data items, has received increased attention in recent years. However, most quantification research has concentrated on developing algorithms for binary and multiclass problems in which the classes are not ordered. We here study the ordinal case, i.e., the case in which a total order is defined on the set of $n > 2$ classes. We give three main contributions to this field. First, we create and make available two datasets for ordinal quantification (OQ) research that overcome the inadequacies of the previously available ones. Second, we experimentally compare the most important OQ algorithms proposed in the literature so far. To this end, we bring together algorithms that are proposed by authors from very different research fields, who were unaware of each other's developments. Third, we propose three OQ algorithms, based on the idea of preventing ordinally implausible estimates through regularization. Our experiments show that these algorithms outperform the existing ones if the ordinal plausibility assumption holds.

Keywords: Quantification · Ordinal classification · Supervised prevalence estimation

1 Introduction

Quantification is a supervised learning task that consists of training a predictor, on a set of labelled data items, that estimates the relative frequencies $p_\sigma(y_i)$ of the classes of interest $\mathcal{Y} = \{y_1, ..., y_n\}$ in a sample σ of unlabelled data items [16]. In other words, a trained *quantifier* must return a *predicted distribution* $\hat{p}_\sigma$ of the unlabelled data items in σ across the classes in $\mathcal{Y}$, where $\hat{p}_\sigma$ must diverge as little as possible from the true, unknown distribution p_σ. Quantification is also known as "learning to quantify", "supervised class prevalence estimation", and "class prior estimation".

Quantification is important in many disciplines, e.g., market research, political science, the social sciences, and epidemiology. By their own nature, these disciplines are only interested in aggregate, as opposed to individual, data. Hence, classifying individual unlabelled instances is usually not a primary goal, while estimating the prevalence values $p_\sigma(y_i)$ of the classes of interest is. For instance, when classifying the tweets about a certain entity, e.g., about a political candidate, as displaying either a Positive or a Negative stance towards the entity, we are usually not interested in the class of a specific tweet, but in the fraction of these tweets that belong to each class [17].

A predicted distribution $\hat{p}_\sigma$ could, in principle, be obtained via the "classify and count" method (CC), i.e., by training a standard classifier, classifying all the unlabelled data items in σ, counting how many of them have been assigned to each class in $\mathcal{Y}$, and normalizing. However, it has been shown that CC delivers poor prevalence estimates, and especially so when the application scenario suffers from *prior probability shift* [22], the (ubiquitous) phenomenon according to which the distribution $p_U(y_i)$ of the *un-labelled* test documents U across the classes is different from the distribution $p_L(y_i)$ of the *labelled* training documents L. As a result, a plethora of quantification methods has been proposed in the literature [4,16,14,17,33] that aims at accurate class prevalence estimations even in the presence of prior probability shift.

The vast majority of the methods proposed so far deals with quantification tasks in which $\mathcal{Y}$ is a plain, unordered set. Very few methods, instead, deal with *ordinal quantification* (OQ), the task of performing quantification on a set of $n > 2$ classes on which a total order "$\prec$" is defined. Ordinal quantification is important, though, because ordinal scales arise in many applications, especially ones involving human judgements. For instance, in a customer satisfaction endeavour, one may want to estimate how a set of reviews of a certain product distributes across the set of classes $\mathcal{Y}$ ={1Star, 2Stars, 3Stars, 4Stars, 5Stars}, while a social scientist might want to find out how inhabitants of a certain region are distributed in terms of their happiness with health services in the area, $\mathcal{Y}$ ={VeryUnhappy, Unhappy, Happy, VeryHappy}.

In this paper, we contribute to the field of OQ in three ways.

First, we develop and publish two datasets for evaluating OQ algorithms, one consisting of textual product reviews and one consisting of telescope observations. Both datasets stem from scenarios in which OQ arises naturally, and they are generated according to a strong, well-tested protocol for the evaluation of quantifiers. The datasets that were previously used for the evaluation of OQ algorithms were not adequate, for reasons we discuss in Sec. 2.

Second, we perform an extensive experimental comparison among the most important OQ algorithms that have been proposed in the literature. This contribution is important because some algorithms have been evaluated on a testbed that was likely inadequate, while some other algorithms have been developed independently of the previous ones, and have thus never been compared to them.

Third, we propose new OQ algorithms, which introduce regularization into existing quantification methods. We experimentally compare our proposals with the existing state of the art and make the corresponding code publicly available[3].

This paper is organized as follows. In Sec. 2, we review past work on OQ. In Sec. 3, we present quantification algorithms, starting with previously proposed ones and then moving to the ones we propose in this work. Sec. 4 is devoted to our experimental evaluation and Sec. 5 concludes.

2 Related work

Quantification, as a task of its own, was first proposed by Forman [16], who observed that some applications of classification only require the estimation of class prevalence values, and that better methods than "classify and count" can be devised for this requirement. Since then, many methods for quantification have been proposed. However, most of these methods tackle the binary and/or multiclass problem with un-ordered classes. While OQ was first discussed in [15] it was not until 2016 that the first true OQ algorithms were developed, the *Ordinal Quantification Tree* (OQT) [10] and *Adjusted Regress and Count* (ARC) [13]. In the same years, the first data challenges that involved OQ were staged [25,30,18]. However, except for OQT and ARC, the participants in these challenges used "classify and count" with highly optimized classifiers, and no true OQ methods; this attitude persisted also in later challenges [39,40], likely due to a general lack of awareness in the scientific community that more accurate methods than "classify and count" exist.

Unfortunately, the data challenges, in which OQT and ARC were evaluated [25,30], tested each quantification method only on a single sample of unlabelled items, which consisted of the entire test set. This evaluation protocol is not adequate for quantification because quantifiers issue predictions for samples of data items, not for individual data items as in classification. Measuring a quantifier's performance on a single sample is thus akin to, and as insufficient as, measuring a classifier's performance on a single data item. As a result, our knowledge of the relative merits of OQT and ARC lacks solidity.

However, even before the previously mentioned developments took place, what we now would call OQ algorithms had been proposed within experimental physics. In this field, we often need to estimate the distribution of a continuous physical quantity. However, a histogram approximation of a continuous distribution is sufficient for many physics-related analyses [6]. This conventional simplification essentially maps the values of a continuous target quantity into a set of classes endowed with a total order, and the problem of estimating the continuous distribution becomes one of OQ. Early on, physicists had termed this problem "unfolding" [5,11], a terminology that prevented quantification researchers from drawing connections between algorithms proposed in the quantification literature and those proposed in the physics literature. In the following, we provide these

[3]Code and supplementary results: https://github.com/mirkobunse/ecml22

connections, to find that regularization techniques proposed within the physics literature are able to improve well-known quantification methods in ordinal settings. We complete the unification of unfolding and quantification methods in [8].

3 Methods

We use the following notation. By $\mathbf{x} \in \mathcal{X}$ we indicate a data item drawn from a domain $\mathcal{X}$, and by $y \in \mathcal{Y}$ we indicate a class drawn from a set of classes $\mathcal{Y} = \{y_1, ..., y_n\}$, also known as a *codeframe*, on which a total order "$\prec$" is defined. The symbol σ denotes a *sample*, i.e., a non-empty set of unlabelled data items in $\mathcal{X}$, while $L \subset \mathcal{X} \times \mathcal{Y}$ denotes a set of labelled data items $(\mathbf{x}, y)$, which we use to train our quantifiers. By $p_\sigma(y)$, we indicate the true prevalence of class y in sample σ, while by $\hat{p}^M_\sigma(y)$, we indicate an estimate of this prevalence, as obtained by a quantification method M that receives σ as an input, where $0 \leq p_\sigma(y), \hat{p}^M_\sigma(y) \leq 1$ and $\sum_{y\in\mathcal{Y}} p_\sigma(y) = \sum_{y\in\mathcal{Y}} \hat{p}^M_\sigma(y) = 1$.

3.1 Non-ordinal quantification methods

We start by introducing some important multiclass quantification methods which do not take ordinality into account. These methods provide the foundation for their ordinal extensions, which we develop in Sec. 3.3.

Classify and Count (CC) [16] is the trivial quantification method, where a "hard" classifier $h : \mathcal{X} \to \mathcal{Y}$ generates predictions for all data items $\mathbf{x} \in \sigma$, and the fraction of predictions is used as a prevalence estimate, i.e.,

$$\hat{p}^{\mathrm{CC}}_\sigma(y_i) = \frac{1}{|\sigma|} \cdot \big|\{\mathbf{x} \in \sigma : h(\mathbf{x}) = y_i\}\big|. \tag{1}$$

In its probabilistic variant, **Probabilistic Classify and Count (PCC)** [4], the hard classifier is replaced by a "soft" classifier $s : \mathcal{X} \to [0, 1]^n$ that returns well-calibrated posterior probabilities $[s(\mathbf{x})]_i \equiv \Pr(y_i|\mathbf{x})$, i.e.,

$$\hat{p}^{\mathrm{PCC}}_\sigma(y_i) = \frac{1}{|\sigma|} \cdot \sum_{\mathbf{x}\in\sigma} [s(\mathbf{x})]_i, \tag{2}$$

where $[\mathbf{z}]_i$ denotes the i-th component of some $\mathbf{z}$, and where $\sum_{i=1}^n [s(\mathbf{x})]_i = 1$.

Adjusted Classify and Count (ACC) [16] and **Probabilistic Adjusted Classify and Count (PACC)** [4] are based on the idea of correcting the $\hat{p}^{\mathrm{CC}}_\sigma$ and $\hat{p}^{\mathrm{PCC}}_\sigma$ estimates by using the (hard or soft) misclassification rates estimated on a validation set V, which coincides with L if k-fold cross-validation is used.

In the multiclass setting, we want to estimate a vector of prevalence values $\mathbf{p} \in \mathbb{R}^n$, where $[\mathbf{p}]_i = p_\sigma(y_i)$. In this case, the adjustment of ACC and PACC amounts to solving for $\mathbf{p}$ the system of linear equations

$$\mathbf{q} = \mathbf{M}\mathbf{p}, \tag{3}$$

where $\mathbf{q} \in \mathbb{R}^n$ is a vector of unadjusted prevalence estimates obtained via CC or PCC, i.e., $[\mathbf{q}]_i^{\text{ACC}} = \hat{p}_\sigma^{\text{CC}}(y_i)$ and $[\mathbf{q}]_i^{\text{PACC}} = \hat{p}_\sigma^{\text{PCC}}(y_i)$, and $\mathbf{M} \in \mathbb{R}^{n \times n}$ is a matrix that relates the ground truth labels to the predictions of the employed classifier, i.e.,

$$\mathbf{M}_{ij}^{\text{ACC}} = \frac{|\{(\mathbf{x}, y) \in V : h(\mathbf{x}) = y_i,\ y = y_j\}|}{|\{(\mathbf{x}, y) \in V : y = y_j\}|}, \tag{4}$$

$$\mathbf{M}_{ij}^{\text{PACC}} = \frac{\sum_{(\mathbf{x},y)\in V : y=y_j} [s(\mathbf{x})]_i}{|\{(\mathbf{x}, y) \in V : y = y_j\}|}. \tag{5}$$

ACC and PACC solve Eq. 3 via the Moore-Penrose pseudo-inverse $\mathbf{M}^\dagger$, i.e.,

$$\hat{\mathbf{p}} = \mathbf{M}^\dagger \mathbf{q}, \tag{6}$$

where $\hat{p}_\sigma^{\text{ACC}}(y_i) \equiv [\hat{\mathbf{p}}]_i$ if Eqs. 1 and 4 are employed, while $\hat{p}_\sigma^{\text{PACC}}(y_i) \equiv [\hat{\mathbf{p}}]_i$ if Eqs. 2 and 5 are employed.

Unlike the true inverse $\mathbf{M}^{-1}$, the pseudo-inverse always exists. If the true inverse exists, the two matrices are identical; if it does not exist, the solution from Equation 6 amounts to a minimum-norm least-squares estimate of $\mathbf{p}$ [23, Theorem 4.1].

EM-based quantification, also known as the **Saerens-Latinne-Decaestecker (SLD)** method [33], follows an expectation maximization approach, which (i) leverages Bayes' theorem in the E-step, and (ii) updates the prevalence estimates in the M-step. Both steps can be combined in the single update rule

$$\hat{p}_\sigma^{(k)}(y_i) = \frac{1}{|\sigma|} \sum_{\mathbf{x}\in\sigma} \frac{\frac{\hat{p}_\sigma^{(k-1)}(y_i)}{\hat{p}_\sigma^{(0)}(y_i)} \cdot [s(\mathbf{x})]_i}{\sum_{j=1}^{n} \frac{\hat{p}_\sigma^{(k-1)}(y_j)}{\hat{p}_\sigma^{(0)}(y_j)} \cdot [s(\mathbf{x})]_j}, \tag{7}$$

which is applied until the estimates converge. $p_\sigma^{(0)}(y)$ is initialized with the class prevalence values of the training set.

3.2 Existing OQ methods from the physics literature

Similar to the adjustment of ACC, experimental physicists have proposed adjustments that solve for $\mathbf{p}$ the system of linear equations from Eq. 3. However, these "unfolding" quantifiers differ from ACC in two regards.

The first aspect is that the hard classifier h of Equations 1 and 4 is often, although not always, replaced by a partition $c : \mathcal{X} \to \{1, \ldots, d\}$ of the feature space, so that

$$\begin{aligned} [\mathbf{q}]_i &= \frac{1}{|\sigma|} \cdot |\{\mathbf{x} \in \sigma : c(\mathbf{x}) = i\}|, \\ \mathbf{M}_{ij} &= \frac{|\{(\mathbf{x}, y) \in V : c(\mathbf{x}) = i,\ y = y_j\}|}{|\{(\mathbf{x}, y) \in V : y = y_j\}|}, \end{aligned} \tag{8}$$

and $\mathbf{M} \in \mathbb{R}^{d \times n}$. Note that by choosing $c = h$ we obtain exactly Equations 1 and 4. Another possible choice for c is to partition the feature space by means of a decision tree; in this case, (i) it typically holds that $d > n$, and (ii) $c(\mathbf{x})$ represents the index of a leaf node [7].

The second difference to ACC is that "unfolding" quantifiers regularize their estimates in order to promote solutions that are the most *plausible* solutions in OQ. Specifically, these methods employ the assumption that neighbouring classes have similar prevalence values; depending on the algorithm, this assumption is encoded in different ways, as we will see in the following paragraphs. This assumption is quite reasonable in OQ because the "smoothness" of the histogram that represents the distribution is arguably *the most important aspect that distinguishes an ordinal distribution from a non-ordinal multiclass distribution*. Without an order of classes, the concept of neighboring classes would even be ill-defined.

Regularized Unfolding (RUN) [6,5] is used by physicists for decades [27,2]. It estimates the vector $\mathbf{p}$ of class prevalence values by minimizing a loss function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ over the estimate $\hat{\mathbf{p}}$; $\mathcal{L}$ consists of two terms, i.e., a negative log-likelihood term to model the error of $\hat{\mathbf{p}}$, and a regularization term to model the plausibility of $\hat{\mathbf{p}}$.

The negative log-likelihood term in $\mathcal{L}$ builds on a Poisson assumption about the distribution of the data. Namely, this term models the counts $[\bar{\mathbf{q}}]_i = |\sigma| \cdot [\mathbf{q}]_i$, which are observed in the sample σ, as being Poisson-distributed with the rates $\lambda_i = \mathbf{M}_i^\top \bar{\mathbf{p}}$. Here, $\mathbf{M}_i$ is the i-th column vector of $\mathbf{M}$ and $[\bar{\mathbf{p}}]_i = |\sigma| \cdot [\hat{\mathbf{p}}]_i$ are the class counts that would be observed under a prevalence estimate $\hat{\mathbf{p}}$.

The second term of $\mathcal{L}$ is a Tikhonov regularization term $\frac{1}{2}\left(\mathbf{C}\mathbf{p}\right)^2$. This term introduces an inductive bias towards solutions which are plausible with respect to ordinality. The Tikhonov matrix $\mathbf{C}$ is chosen in such a way that term $\frac{1}{2}\left(\mathbf{C}\mathbf{p}\right)^2$ measures the smoothness of the histogram that represents the distribution, i.e.,

$$\frac{1}{2}\left(\mathbf{C}\mathbf{p}\right)^2 = \frac{1}{2}\sum_{i=2}^{n-1}\left(-[\mathbf{p}]_{i-1} + 2[\mathbf{p}]_i - [\mathbf{p}]_{i+1}\right)^2. \tag{9}$$

Combining the likelihood term and the regularization term, the loss function of RUN is given by

$$\mathcal{L}(\hat{\mathbf{p}};\ \mathbf{M}, \mathbf{q}, \tau, \mathbf{C}) = \sum_{i=1}^{d}\left(\mathbf{M}_i^\top \bar{\mathbf{p}} - [\bar{\mathbf{q}}]_i \cdot \ln(\mathbf{M}_i^\top \bar{\mathbf{p}})\right) + \frac{\tau}{2}\left(\mathbf{C}\hat{\mathbf{p}}\right)^2 \tag{10}$$

and an estimate $\hat{\mathbf{p}}$ is chosen by minimizing $\mathcal{L}$ numerically over $\hat{\mathbf{p}}$. Here, $\tau \geq 0$ is a hyperparameter which controls the impact of the regularization.

Iterative Bayesian Unfolding (IBU) [12,11] is still popular today [1,24]. This method revolves around an expectation maximization approach with Bayes'

theorem, and thus has a common foundation with the SLD method. The E-step and the M-step of IBU can be written as the single, combined update rule

$$\hat{p}_\sigma^{(k)}(y_i) = \sum_{j=1}^{d} \frac{\mathbf{M}_{ij} \cdot \hat{p}_\sigma^{(k-1)}(y_i)}{\sum_{l=1}^{n} \mathbf{M}_{lj} \cdot \hat{p}_\sigma^{(k-1)}(y_l)} [\mathbf{q}]_i. \tag{11}$$

One difference between IBU and SLD is that $\mathbf{q}$ and $\mathbf{M}$ are defined via counts of hard assignments to partitions $c(\mathbf{x})$ (see Eq. 8), while SLD is defined over individual soft predictions $s(\mathbf{x})$ (see Eq. 7).

Another difference between IBU and SLD is regularization. In order to promote solutions which are ordinally plausible, IBU smooths each intermediate estimate $\hat{\mathbf{p}}^{(k)}$ by fitting a low-order polynomial to $\hat{\mathbf{p}}^{(k)}$. A linear interpolation between $\hat{\mathbf{p}}^{(k)}$ and this polynomial is then used as the prior of the next iteration in order to reduce the differences between neighbouring prevalence estimates. The interpolation factor is a hyperparameter of IBU through which the degree of regularization is controlled.

Other methods from the physics literature, which go under the name of "unfolding", are based on similar concepts as RUN and IBU. We focus on these two methods due to their long-standing popularity within physics research. In fact, they are among the first methods that have been proposed in this field, and are still widely adopted today, in astro-particle physics [27,2], high-energy physics [1], and more recently in quantum computing [24]. Moreover, RUN and IBU already cover the most important aspects of unfolding methods with respect to ordinal quantification.

Several other unfolding methods are similar to RUN. The method proposed in [19], for instance, employs the same regularization as RUN, but assumes different Poisson rates, which are simplifications of the rates that RUN uses; in preliminary experiments, here omitted for the sake of conciseness, we have found this simplification to typically deliver less accurate results than RUN. Two other methods [35,36] employ the same simplification as [19] but regularize differently. To this end, [35] regularizes with respect to the deviation from a prior, instead of regularizing with respect to ordinal plausibility; we thus do not perceive this method as a true OQ method. [36] adds to the RUN regularization a second term which enforces prevalence estimates that sum up to one; we use a RUN implementation which already solves this issue through a positivity constraint and normalization. Another line of work evolves around the algorithm of [32] and its extensions [9]. We perceive this algorithm to lie outside the scope of OQ because it does not address the order of classes, like the other "unfolding" methods do. Moreover, the algorithm was shown to exhibit a performance comparable to, but not better than RUN and IBU [9].

3.3 New ordinal variants of ACC, PACC, and SLD

In the following, we develop algorithms which extend ACC, PACC, and SLD with the regularizers from RUN and IBU. Through these extensions, we obtain

o-ACC, o-PACC, and o-SLD, the OQ counterparts of these well-known non-ordinal quantification algorithms. In doing so, since we employ the regularizers but not any other aspect of RUN and IBU, we preserve the general characteristics of ACC, PACC, and SLD. In particular, our methods continue to work with classifier predictions, i.e., we do not employ the categorical feature representation from Eq. 8, which RUN and IBU employ, and we do not use the Poisson assumption of RUN. Therefore, our extensions are "minimal", in the sense of being directly addressed to ordinality and not introducing any undesired side effects in the original methods.

o-ACC and o-PACC, our ordinal extensions to ACC and PACC, build on the finding reported in [23, Theorem 4.1], which states that the solution from Eq. 6 corresponds to a minimum-norm least-squares solution. Namely, among all least-squares solutions $\hat{\mathbf{p}}^{\mathrm{LSq}} = \arg\min_{\mathbf{p}} \|\mathbf{q} - \mathbf{M}\mathbf{p}\|_2^2$, which by themselves do not need to be unique, the solution to Eq. 6 is the unique one that also minimizes the quadratic norm $\|\mathbf{p}\|_2^2$. Therefore, Eq. 6 is conceptually similar, although not necessarily equal, to a regularized estimate

$$\hat{\mathbf{p}}' = \arg\min_{\mathbf{p}} \|\mathbf{q} - \mathbf{M}\mathbf{p}\|_2^2 + \frac{\tau}{2}\|\mathbf{p}\|_2^2, \tag{12}$$

which employs the quadratic norm for regularization. In particular, both Equations 6 and 12 simultaneously minimize a least-squares objective and the norm of their candidate solutions. Note that the regularization function herein is, unlike the regularization from RUN, unrelated to the ordinal nature of the classes.

To obtain the true OQ methods o-ACC and o-PACC, we replace the minimum-norm regularization in Eq. 12 with the regularization term of RUN (see Eq. 9). Through this replacement, we minimize the same objective function as ACC and PACC, i.e., a least-squares objective, but regularize towards solutions that we deem more plausible for OQ. The prevalence estimate is

$$\hat{\mathbf{p}}^{\mathrm{o}} = \arg\min_{\mathbf{p}} \|\mathbf{q} - \mathbf{M}\mathbf{p}\|_2^2 + \frac{\tau}{2}\left(\mathbf{C}\mathbf{p}\right)^2, \tag{13}$$

the minimizer of which can be found through numerical optimization, e.g., through the BFGS optimization technique [26]. The o-ACC variant emerges from plugging in Eqs. 1 and 4 for $\mathbf{q}$ and $\mathbf{M}$, while the o-PACC variant emerges from plugging in Eqs. 2 and 5.

o-SLD leverages the ordinal regularization of IBU in SLD. Namely, our method does not use the latest estimate directly as the prior of the next iteration, but a smoothed version of this estimate. To this end, we fit a low-order polynomial to each intermediate estimate $\hat{\mathbf{p}}^{(k)}$ and use a linear interpolation between this polynomial and $\hat{\mathbf{p}}^{(k)}$ as the prior of the next iteration. Like in IBU, we consider the interpolation factor as a hyperparameter through which the strength of this regularization is controlled.

4 Experiments

The goal of our experiments is to uncover the relative merits of OQ methods that come from different fields. We pursue this goal by carrying out a thorough comparison of these methods on representative OQ data sets.

4.1 Evaluation measures

The main evaluation measure we use in this paper is the *Normalized Match Distance* (NMD), defined by [34] as

$$\mathrm{NMD}(p, \hat{p}) = \frac{1}{n-1}\,\mathrm{MD}(p, \hat{p}), \tag{14}$$

where $\frac{1}{n-1}$ is just a normalization factor that allows NMD to range between 0 (best) and 1 (worst). Here MD is the *Match Distance* [38], defined as

$$\mathrm{MD}(p, \hat{p}) = \sum_{i=1}^{n-1} d(y_i, y_{i+1}) \cdot |\hat{P}(y_i) - P(y_i)|, \tag{15}$$

where $d(y_i, y_{i+1})$ is the "distance" between consecutive classes y_i and y_{i+1}, i.e., the cost we incur in assigning to y_i a probability mass that we should instead assign to y_{i+1}, or vice versa. Here, we assume $d(y_i, y_{i+1}) = 1$ for all $i \in \{1, ..., n-1\}$ and $P(y_i) = \sum_{j=1}^{i} p(y_j)$ is the cumulative distribution of p.

MD is a special case of the *Earth Mover's Distance* (EMD) [31], a widely used measure in OQ evaluation [15,25,30,10,9]. Since MD and EMD differ only by a fixed normalization factor, our experiments perfectly follow the tradition in OQ evaluation.

Another proposed measure for evaluating the quality of OQ estimates is the *Root Normalized Order-aware Divergence* (RNOD) [34]. We include a definition of, and an evaluation in terms of, RNOD in the supplementary material[3], where we find that RNOD and NMD consistently lead to the same conclusions.

To obtain an overall score for a quantification method on a dataset, we apply this method to each test sample σ. The resulting prevalence estimates are then compared to the ground-truth prevalence values, which yields one NMD (or RNOD) value for each sample. The final score of the method is the average of these values, i.e., the average NMD (or RNOD) across all samples in the test set. We test for statistically significant differences between quantification methods in terms of a paired Wilcoxon signed-rank test.

4.2 Datasets and preprocessing

We conduct our experiments on two large datasets that we have generated for the purpose of this work, and that we make available to the scientific community[3]. The first dataset, named Amazon-OQ-BK, consists of product reviews labelled according to customer's judgments of quality, i.e., 1Star to 5Stars. The

second dataset, FACT-OQ, consists of telescope observations labelled by one of 12 totally ordered classes. Hence, these data sets originate in practically relevant and diverse applications of OQ. Each of these data sets consists of a training set, multiple validation samples, and multiple test samples, which are extracted from the original data source according to two extraction protocols that are well suited for OQ.

The data sampling protocol. We start by dividing a set of labelled data items into a training set L, a pool of validation (i.e., development) items, and a pool of test items. These three sets are disjoint from each other, and each of them is obtained through stratified sampling from the original data sources.

From each of the pools, we separately extract samples for quantifier evaluation. This extraction follows the so-called *Artificial Prevalence Protocol* (APP), by now a standard extraction protocol in quantifier evaluation (see, e.g., [16]). This protocol generates each sample in two steps. The first step consists of generating a vector $\mathbf{p}_\sigma$ of class prevalence values. Following [14], we generate this vector by drawing uniformly at random from the set of all legitimate prevalence vectors by using the Kraemer algorithm [37], which (differently from other naive algorithms) ensures that all prevalence values in the unit $(n-1)$ simplex are picked with equal probability. Since each $\mathbf{p}_\sigma$ can be, and typically is, different from the training set prevalences, this approach covers the entire space of prior probability shifts. The second step consists of drawing from the pool of data, be it our validation pool or our test pool, a fixed-size sample σ of data items which obeys the class prevalence values of $\mathbf{p}_\sigma$. The result is a set of samples characterized by uniformly distributed vectors of prevalence values, which give rise to varying levels of prior probability shift. We obtain one such set of samples from the validation pool and another set from the test pool.

For our two datasets, (i) we set the size of the training set to 20,000 data items, (ii) we have each (validation or test) sample consist of 1000 data items, (iii) we have the validation set consist of 1000 such samples, and (iv) we have the test set consist of 5000 such samples. For AMAZON-OQ-BK, a data item corresponds to a single product review, while for FACT-OQ, a data item corresponds to a single telescope recording.

All items in the pool are replaced after the generation of each sample, so that no sample contains duplicate items but samples from the same pool are not necessarily disjoint. Note, however, that our initial split into a training set, a validation pool, and a test pool ensures that each validation sample is disjoint from each test sample, and that the training set is disjoint from all other samples.

Partitioning samples in terms of ordinal plausibility. In APP, all class prevalence vectors are sampled with the same probability, disregarding of their "plausibility", in the sense of being likely to appear in the practice of OQ. For instance, $\mathbf{p}_1 = (.0, .5, .0, .5, .0)$ and $\mathbf{p}_2 = (.2, .1, .0, .3, .4)$ have the same chances to be generated within APP, despite the fact that $\mathbf{p}_1$ seems much less likely

to show up than $\mathbf{p}_2$ in a real OQ application. Indeed, a vector such as $\mathbf{p}_2$ is extremely likely in the realm of product reviews.

We counteract this shortcoming of APP by using APP-OQ, a protocol similar to APP but for the fact that only samples "plausible" in the context of OQ are considered. Namely, in APP-OQ, we retain only the 20% most plausible samples generated by APP. Hence, we perform hyperparameter optimization on the selected 20% validation samples, and perform the evaluation on the selected 20% test samples. We always report the results of both APP and APP-OQ side by side, so as to allow drawing conclusions concerning the OQ-related merits of the different quantification methods.

Motivated by our experience in sentiment quantification and unfolding, we use "smoothness" as an indicator of plausibility. We measure smoothness by applying Eq. 9 to the class prevalence vector $\mathbf{p}$ of each sample, so that the most plausible samples are those with the smallest value of $\frac{1}{2}(\mathbf{Cp})^2$. We recognize that this measure can only be a first step towards assessing the plausibility of prevalence vectors in OQ because plausibility necessarily depends on the use case and on the expected number of data items in each sample.

The Amazon-OQ-BK dataset is extracted from an existing dataset[4] of 233.1M English-language Amazon product reviews, made available by [21]; here, a data item corresponds to a single product review. As the labels of the reviews, we use their "stars" ratings, and our codeframe is thus $\mathcal{Y}$ ={1Star, 2Stars, 3Stars, 4Stars, 5Stars}, which represents a sentiment quantification task [15].

We restrict our attention to reviews from the Books domain. We then remove (a) all reviews shorter than 200 characters because recognizing sentiment from shorter reviews may be nearly impossible in some cases, and (b) all reviews that have not been recognized as "useful" by any users because many reviews never recognized as "useful" may contain comments, say, on Amazon's speed of delivery, and not on the product itself.

We convert the documents into vectors by using the RoBERTa transformer [20] from the Hugging Face hub[5]. To this aim, we truncate the documents to the first 256 tokens, and fine-tune RoBERTa via prompt learning for a maximum of 5 epochs on our training data, thus taking the model parameters from the epoch which yields the smallest validation loss as monitored on 1000 held-out documents randomly sampled from the training set in a stratified way. For training, we set the learning rate to $2e^{-5}$, the weight decay to 0.01, and the batch size to 16, leaving the other hyperparameters at their default values. For each document, we generate features by first applying a forward pass over the fine-tuned network, and then averaging the embeddings produced for the special token [CLS] across all the 12 layers of RoBERTa. In our initial experiments, this latter approach yielded slightly better results than using the [CLS] embedding of the last layer alone. The embedding size of RoBERTa, and hence the number of dimensions of our vectors, amounts to 768.

[4] http://jmcauley.ucsd.edu/data/amazon/links.html
[5] https://huggingface.co/docs/transformers/model_doc/roberta

We make the Amazon-OQ-BK dataset publicly available[3], both in its raw textual form and in its processed vector form.

The Fact-OQ dataset is extracted from the open dataset[6] of the FACT telescope [3]; here, a data item corresponds to a single telescope recording. We represent each data item in terms of the 20 dense features that are extracted by the standard processing pipeline[7] of the telescope. Each of the 1,851,297 recordings is labelled with the energy of the corresponding astro-particle, and our goal is to estimate the distribution of these energy labels via OQ. While the energy labels are originally continuous, astro-particle physicists have established a common practice of dividing the range of energy values into ordinal classes, as argued in Sec. 3.2. Based on discussions with astro-particle physicists, we divide the range of continuous energy values into an ordered set of 12 classes.

4.3 Results with ordinal classifiers

In our first experiment, we investigate whether OQ can be solved by non-ordinal quantification methods that embed ordinal classifiers. To this end, we compare the use of a standard multiclass logistic regression (LR) with the use of several ordinal variants of LR. In general, we have found that LR models, trained on the deep RoBERTa embedding of the Amazon-OQ-BK dataset, are extremely powerful models in terms of quantification performance. Therefore, approaching OQ with ordinal LR variants embedded in non-ordinal quantifiers could be a straightforward solution worth investigating.

The ordinal LR variants we test are the "All Threshold" variant (OLR-AT) and the "Immediate-Threshold variant" (OLR-IT) of [29]. In addition, we try two ordinal classification methods based on discretizing the outputs generated by regression models [28]; the first is based on *Ridge Regression* (ORidge) while the second, called *Least Absolute Deviation* (LAD), is based on linear SVMs.

Tab. 1 reports the results of this experiment, using the non-ordinal quantifiers of Sec. 3.1. The fact that the best results are almost always obtained by using, as the embedded classifier, non-ordinal LR shows that, in order to deliver accurate estimates of class prevalence values in the ordinal case, it is not sufficient to equip a multiclass quantifier with an ordinal classifier. Moreover, the poor results of PCC, PACC, and SLD, the three methods that make use of posterior probabilities, suggest that the quality of the posterior probabilities returned by the ordinal classifiers may be sub-optimal.

Overall, these results suggest that, in order to tackle OQ, we cannot simply rely on ordinal classifiers embedded in non-ordinal quantification methods. Instead, we need "real" OQ methods.

[6] https://factdata.app.tu-dortmund.de/
[7] https://github.com/fact-project/open_crab_sample_analysis/

Table 1. Performance of classifiers in terms of average NMD (lower is better) in the Amazon-OQ-BK dataset. **Boldface** indicates the best classifier for each quantification method, or a classifier not significantly different from the best one in terms of a paired Wilcoxon signed-rank test at a confidence level of $p = 0.01$. For LR we present standard deviations, while for all other classifiers we show the average deterioration in NMD with respect to LR. PCC, PACC, and SLD require a soft classifier, which means that ORidge and LAD cannot be embedded in these methods.

	CC	PCC	ACC	PACC	SLD
LR	**.0526** ±.0190	**.0629** ±.0215	.0247 ±.0096	**.0206** ±.0080	**.0174** ±.0068
OLR-AT	.0527 (+0.2%)	.0657 (+4.4%)	**.0237** (−4.4%)	.0219 (+6.5%)	.0210 (+20.5%)
OLR-IT	**.0526** (+0.0%)	.0695 (+10.4%)	.0256 (+3.6%)	.0215 (+4.5%)	.0648 (+271.8%)
ORidge	.0550 (+4.5%)	—	.0244 (−1.6%)	—	—
LAD	**.0527** (+0.3%)	—	**.0240** (−3.1%)	—	—

4.4 Results of the quantifier comparison

In our main experiment, we compare our proposed methods o-ACC, o-PACC, and o-SLD with several baselines, i.e., (i) the existing OQ methods OQT [10] and ARC [13], which we further detail in the supplementary material[3], (ii) the "unfolding" OQ methods IBU and RUN (see Sec. 3.2), and (iii) the non-ordinal methods CC, PCC, ACC, PACC, and SLD. We compare these methods on the Amazon-OQ-BK and Fact-OQ datasets, and under the APP and APP-OQ protocols.

Each method is allowed to tune the hyperparameters of its embedded classifier, using the samples of the validation set. We use logistic regression on Amazon-OQ-BK and probability-calibrated decision trees on Fact-OQ; this choice of classifiers is motivated by common practice in the fields where these data sets originate, and from our own experience that these classifiers work well on the respective type of data. After the hyperparameters of the classifier are optimized, we apply each method to the samples of the test set.

The results of this experiment are summarized in Tab. 2. These results show that our proposed methods outperform the competition on both data sets if the ordinal APP-OQ protocol is employed. More specifically, o-SLD is the best method on Amazon-OQ-BK while o-PACC is the best method on Fact-OQ. Moreover, o-SLD is consistently better or equal to SLD, o-ACC is consistently better or equal to ACC, and o-PACC is consistently better or equal to PACC, also in the standard APP protocol, where smoothness is not imposed.

Using RNOD as an alternative error measure confirms these conclusions, while experiments carried out using additional datasets and using TFIDF as an alternative vectorial representation in Amazon-OQ-BK, even reinforce these conclusions. We provide these results in the supplementary material[3].

Table 2. Average performance in terms of NMD (lower is better). For each data set (Amazon-OQ-BK and Fact-OQ), we present the results of the two protocols APP and APP-OQ. The best performance in each column is highlighted in **boldface**. According to a Wilcoxon signed rank test with $p = 0.01$, all other methods are statistically significantly different from the best method.

method	Amazon-OQ-BK		Fact-OQ	
	APP	APP-OQ	APP	APP-OQ
CC	$.0526 \pm .019$	$.0344 \pm .013$	$.0534 \pm .012$	$.0494 \pm .011$
PCC	$.0629 \pm .022$	$.0440 \pm .017$	$.0651 \pm .017$	$.0621 \pm .017$
ACC	$.0229 \pm .009$	$.0193 \pm .007$	$.0582 \pm .028$	$.0575 \pm .028$
PACC	$.0209 \pm .008$	$.0176 \pm .007$	$.0791 \pm .048$	$.0816 \pm .049$
SLD	$\mathbf{.0172 \pm .007}$	$.0154 \pm .006$	$.0373 \pm .010$	$.0355 \pm .009$
OQT	$.0775 \pm .026$	$.0587 \pm .027$	$.0746 \pm .019$	$.0731 \pm .020$
ARC	$.0641 \pm .023$	$.0477 \pm .015$	$.0566 \pm .014$	$.0568 \pm .016$
IBU	$.0253 \pm .010$	$.0197 \pm .007$	$\mathbf{.0213 \pm .005}$	$.0187 \pm .004$
RUN	$.0252 \pm .010$	$.0198 \pm .007$	$.0222 \pm .006$	$.0194 \pm .005$
o-ACC	$.0229 \pm .009$	$.0188 \pm .007$	$.0274 \pm .007$	$.0230 \pm .006$
o-PACC	$.0209 \pm .008$	$.0174 \pm .007$	$.0230 \pm .006$	$\mathbf{.0178 \pm .004}$
o-SLD	$.0173 \pm .007$	$\mathbf{.0152 \pm .006}$	$.0327 \pm .008$	$.0289 \pm .007$

5 Conclusion

We have carried out a thorough investigation of ordinal quantification, which includes (i) making available two datasets for OQ, generated according to the strong extraction protocols APP and APP-OQ, which overcome the limitations of existing OQ datasets, (ii) showing that OQ cannot be profitably tackled by simply embedding ordinal classifiers into non-ordinal quantification methods, (iii) proposing three OQ methods (o-ACC, o-PACC, and o-SLD) that combine intuitions from existing, non-ordinal quantification methods and existing, physics-inspired "unfolding" methods, and (iv) experimentally comparing our newly proposed OQ methods with existing non-ordinal quantification methods, ordinal quantification methods, and "unfolding" methods, which we have shown to be OQ methods under a different name. Our newly proposed methods outperform the competition when tested on "ordinally plausible" test data. Our supplementary material[3] confirms these results with evaluations under a different error measure and with additional experiments that we have carried out on different datasets and using a different, vectorial representation of the text data.

At the heart of the success of our newly proposed method lies regularization, which is motivated by the assumption that typical OQ class prevalence vectors are smooth. In future work, we plan to attempt using regularization for turning other non-ordinal quantification methods into ordinal ones.

Acknowledgments The work by M.B., A.M., and F.S. has been supported by the European Union's Horizon 2020 research and innovation programme under

grant agreement No. 871042 (SoBigData++). M.B. and M.S. have further been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Data Analysis", project C3, https://sfb876.tu-dortmund.de. A.M. and F.S. have further been supported by the AI4Media project, funded by the European Commission (Grant 951911) under the H2020 Programme ICT-48-2020. The authors' opinions do not necessarily reflect those of the European Commission.

References

1. Aad, G., Abbott, B., Abbott, D.C., et al.: Measurements of the inclusive and differential production cross sections of a top-quark–antiquark pair in association with a Z boson at $\sqrt{s} = 13$ TeV with the ATLAS detector. Europ. Phys. J. C **81**(8) (2021)
2. Aartsen, M.G., Ackermann, M., Adams, J., et al.: Measurement of the ν_μ energy spectrum with IceCube-79. Europ. Phys. J. C **77**(10) (2017)
3. Anderhub, H., Backes, M., Biland, A., et al.: Design and operation of FACT, the first G-APD Cherenkov telescope. J. Inst. **8**(06) (2013)
4. Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Quantification via probability estimators. In: Int. Conf. on Data Mining (2010)
5. Blobel, V.: Unfolding methods in high-energy physics experiments. Tech. Rep. DESY-84-118, CERN, Geneva, CH (1985)
6. Blobel, V.: An unfolding method for high-energy physics experiments. In: Adv. Stat. Techn. Part. Phys. pp. 258–267. Durham, UK (2002)
7. Börner, M., Hoinka, T., Meier, M., et al.: Measurement/simulation mismatches and multivariate data discretization in the machine learning era. In: Conf. Astron. Data Anal. Softw. Syst. pp. 431–434 (2017)
8. Bunse, M.: Unification of algorithms for quantification and unfolding. In: Workshop on Mach. Learn. for Astropart. Phys. and Astron. Gesellschaft für Informatik e.V. (2022), to appear
9. Bunse, M., Piatkowski, N., Morik, K., Ruhe, T., Rhode, W.: Unification of deconvolution algorithms for Cherenkov astronomy. In: Data Sci. and Adv. Anal. pp. 21–30 (2018)
10. Da San Martino, G., Gao, W., Sebastiani, F.: Ordinal text quantification. In: Int. ACM SIGIR Conf. on Res. and Dev. in Inf. Retr. pp. 937–940 (2016)
11. D'Agostini, G.: A multidimensional unfolding method based on Bayes' theorem. Nucl. Instr. Meth. Phys. Res.: Sect. A **362**(2-3), 487–498 (1995)
12. D'Agostini, G.: Improved iterative Bayesian unfolding (2010), arXiv:1010.0632
13. Esuli, A.: ISTI-CNR at SemEval-2016 Task 4: Quantification on an ordinal scale. In: Int. Worksh. on Semantic Eval. pp. 92–95 (2016)
14. Esuli, A., Moreo, A., Sebastiani, F.: LeQua@CLEF2022: Learning to Quantify. In: Europ. Conf. on Inf. Retrieval. pp. 374–381 (2022)
15. Esuli, A., Sebastiani, F.: Sentiment quantification. IEEE Intell. Syst. **25**(4), 72–75 (2010)
16. Forman, G.: Counting positives accurately despite inaccurate classification. In: Europ. Conf. Mach. Learn. pp. 564–575 (2005)
17. Gao, W., Sebastiani, F.: From classification to quantification in tweet sentiment analysis. Social Netw. Anal. and Mining **6**(19), 1–22 (2016)

18. Higashinaka, R., Funakoshi, K., Inaba, M., Tsunomori, Y., Takahashi, T., Kaji, N.: Overview of the 3rd Dialogue Breakdown Detection challenge. In: Dialog Syst. Techn. Challenge (2017)
19. Hoecker, A., Kartvelishvili, V.: SVD approach to data unfolding. Nucl. Instr. Meth. Phys. Res.: Sect. A **372**(3), 469–481 (1996)
20. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach (2019), arXiv:1907.11692
21. McAuley, J.J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: Int. ACM SIGIR Conf. on Res. and Dev. in Inf. Retr. pp. 43–52 (2015)
22. Moreno-Torres, J.G., Raeder, T., Alaíz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recogn. **45**(1), 521–530 (2012)
23. Mueller, J.L., Siltanen, S.: Linear and nonlinear inverse problems with practical applications. SIAM (2012)
24. Nachman, B., Urbanek, M., de Jong, W.A., Bauer, C.W.: Unfolding quantum computer readout noise. npj Quant. Inf. **6**(1) (2020)
25. Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., Stoyanov, V.: SemEval-2016 Task 4: Sentiment analysis in Twitter. In: Int. Worksh. on Semantic Eval. pp. 1–18 (2016)
26. Nocedal, J., Wright, S.J.: Numerical optimization. Springer, Cham, CH (2006)
27. Nöthe, M., Adam, J., Ahnen, M.L., et al.: FACT – performance of the first Cherenkov telescope observing with SiPMs. In: Int. Cosmic Ray Conf. (2018)
28. Pedregosa, F., Bach, F., Gramfort, A.: On the consistency of ordinal regression methods. J. Mach. Learn. Res. **18**, 55:1–55:35 (2017)
29. Rennie, J.D., Srebro, N.: Loss functions for preference levels: Regression with discrete ordered labels. In: IJCAI 2005 Worksh. on Adv. in Pref. Handling (2005)
30. Rosenthal, S., Farra, N., Nakov, P.: SemEval-2017 Task 4: Sentiment analysis in Twitter. In: Int. Worksh. on Semantic Eval. pp. 502–518 (2017)
31. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Int. Conf. Comp. Vision. pp. 59–66 (1998)
32. Ruhe, T., Schmitz, M., Voigt, T., Wornowizki, M.: DSEA: A data mining approach to unfolding. In: Int. Cosmic Ray Conf. pp. 3354–3357 (2013)
33. Saerens, M., Latinne, P., Decaestecker, C.: Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. Neur. Comp. **14**(1), 21–41 (2002)
34. Sakai, T.: Comparing two binned probability distributions for information access evaluation. In: Int. ACM SIGIR Conf. on Res. and Dev. in Inf. Retr. pp. 1073–1076 (2018)
35. Schmelling, M.: The method of reduced cross-entropy: A general approach to unfold probability distributions. Nucl. Instr. Meth. Phys. Res.: Sect. A **340**(2), 400–412 (1994)
36. Schmitt, S.: TUnfold, an algorithm for correcting migration effects in high energy physics. J. Inst. **7**(10) (2012)
37. Smith, N.A., Tromble, R.W.: Sampling uniformly from the unit simplex. Tech. rep., Johns Hopkins University (2004)
38. Werman, M., Peleg, S., Rosenfeld, A.: A distance metric for multidimensional histograms. Comp. Vis., Graph., Image Proc. **32**, 328–336 (1985)
39. Zeng, Z., Kato, S., Sakai, T.: Overview of the NTCIR-14 Short Text Conversation task: Dialogue Quality and Nugget Detection subtasks. In: NTCIR (2019)

40. Zeng, Z., Kato, S., Sakai, T., Kang, I.: Overview of the NTCIR-15 Dialogue Evaluation task (DialEval-1). In: NTCIR (2020)